Synergetic Image Recognition with Applications to Pose Estimation

by
Trevor Hogg
B.Sc.(Hons I), Sydney, 1992

Submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

THE UNIVERSITY OF TASMANIA
APRIL 1998
School of Electrical Engineering & Computer Science

Declaration of Originality

I, Trevor Hogg, state that this dissertation contains no material which has been accepted for a degree or diploma by the University of Tasmania, or any other institution. To the best of my knowledge, no material previously published or written by another person is contained in the text of the dissertation, except where due acknowledgement is given.

Trevor Hogg

Authority of Access

Abstract

Synergetics is the study of systems in which individual sub-systems act co-operatively. In particular, we are interested as to how this co-operation on a microscopic scale can lead to the formation of macroscopic spatial structures or patterns.

Synergetic pattern recognition is a non-conventional form of pattern recognition which is modelled on these synergetic pattern formation systems. Indeed the paradigm of synergetic pattern recognition states that pattern recognition is a type of pattern formation.

This thesis is split into two parts. In Part A we introduce synergetic pattern recognition. A review of the current range of synergetic pattern recognition systems leads to a recognition of two major weaknesses in the current crop of synergetic pattern recognition algorithms, so we introduce a number of generalised pattern formation models which extend the capabilities of synergetic pattern recognition.

We also investigate the concept of synergetic learning whereby learning is considered as a type of pattern formation. During a review of the current approach to this task we recognise a number of problems and solve them with an important observation concerning the dependence between variables. The new learning algorithm which results is a significant improvement over the current approach.

In Part B of this work, we discuss the possible application of synergetic pattern recognition to the task of view-based pose estimation, which is the challenge of estimating the angles at which an image has been rotated, without reference to a full model of the object.

Synergetic pattern recognition has not been used to solve this problem previously, so following a review of the view-based pose estimation literature, we introduce two new approaches to pose estimation based on synergetic pattern recognition. Comparison of the results found by the various techniques on a standard dataset reveal a number of common issues, foremost among which is the need to make a compromise between the precision of a system's estimate and the amount of time taken to produce the estimate.

This observation leads to the most exciting finding of this work. We describe a new approach, not just to view-based pose estimation, but to the general field of pattern analysis, which we call *explicit inversion*. The reason for the name is that we have replaced the problem of numerically inverting a high-dimensional, unknown equation with analytically inverting a known, low-dimensional equation. Comparison with other approaches shows that this new approach yields comparable accuracy to current algorithms but with a dramatic reduction in calculation times. For datasets used in this dissertation, the increase in speed was between one to two orders of

magnitude.

We then apply our algorithms to the real-world task of tracking the pose of an aircraft. While a number of attempts have been made previously, this is the only approach which shows promise of being able to track aircraft pose in real-time.

Acknowledgements

I write these acknowledgements in the sure and certain knowledge that for most of you, this is as far into my dissertation as you're likely to read. So they'd better be good.

First and foremost I'd like to thank my supervisors, David and Habib, for their extraodinary efforts. David's boundless enthusiasm for his work, his often crazy ideas and his belief in me (one of the aforementioned crazy ideas), have been inspirational. He has taught me many things, but try as he might, he'll never teach me to like country music.

To Habib, my thanks for rescuing me when the politics up North exploded on me. Without your help, the PhD would have died then and there. As well as his able academic support and guidance, Habib has taught me that when choosing a supervisor, one should choose a great cook. Somehow when one's stomach is in heaven, the mind tends to function at its best. Thanks to Habib and Maida (not to forget Sarri) for their hospitality, especially on cold Hobart nights.

My family have offered me very real support throughout the three years, especially in the first year when I needed it most. You were always there to listen to me. You looked excited when I was excited. And you told me to stop being depressed when I was depressed. For this and everything else, many thanks.

Many allegiances were forged and broken during the time I spent on this work, but the CSIRO remained behind me constantly. For this I would like to thank *everyone* at CSIRO. In particular, my thanks goes to Dennis Cooper, whose generous financial help made many things possible, and to Geoff Poulton, who ensured that I had an environment in which the work could progress, digress and regress.

Thanks also to the staff of the University of Tasmania for their support, particuarly to Thong Nguyen, Judy Bonsey, and to the fabulous people at the Research Office.

To my occasional office-mates; Mal, John, Quang, Fu and Niki, thanks for letting me talk to you incessantly when my computer refused to listen to me anymore. To my flatmates; Louise, Diane and Deborah, thank you one and all for listening to me when my office-mates had had enough. I could not have asked for better flatmates.

There are many other people I'd like to thank; Ray Johnson, Moya Ward, Rob Gill, Glenn Geers, Margot Nichols, Mike Dadd, Jeanne Young, John Kot, Joshua van Kleef, Mary Myerscough, Peter Buchen, Iven Mareels, Peter Bartlett, Andreas Daffertshofer, Rosemarie Bund, the gang from Thursday tennis and, of course, the irrepressible volleyball players.

A vote of thanks goes to all those people who were brave enough to ask me what I was studying *exactly*. An even bigger vote of thanks to those who were foolish enough

to stay around for the answer.

And finally two special thank you notes. Thanks to Jacqui, without whom I would not have started this work. And thanks to Georgina, without whom I would have finished considerably earlier.

Original Contributions

Chapter 2

- Review of synergetic pattern recognition.

Chapter 3

- Analysis of generalised pattern formation model leading to a generalised linear synergetic pattern algorithm.
- Design of deterministic, non-iterative training scheme which is independent of arbitrary user-defined parameters.
- Introduction of a parallel version of the new synergetic algorithm, which provides extra flexibility.

Chapter 4

- Design of an extended pattern formation model which allows for both pattern recognition and rejection at user-defined levels.

Chapter 5

- Critical analysis of the current approach to unsupervised synergetic learning.
- Design and testing of a practical unsupervised synergetic learning system.

Chapter 7

- Design of synergetic warping based pose estimation.
- Design of synergetic prototype warping based pose estimation.

Chapter 8

- Design of synergetic interpolation based pose estimation.
- Creation of code for a gradient-descent based minimisation from a point in an arbitrarily-dimensioned space to a non-uniform rational b-spline surface of the same dimension, parameterised by an arbitrary number of dimensions.

Chapter 9

- Construction of a standard framework for scalar-product based feature extractors.
- Recognition of equivalence of MELT and Optimal Linear Identification Mapping.
- Design of separable, analytically invertible feature extractor mappings for qualitative and quantitative image analysis.

Chapter 10

- Application of view-based pose estimation to multi-dimensional pose estimation.
- Design of pose tracking system.

- Construction of a standard framework for scalar-product based feature extractors.
- Recognition of equivalence of MELT and Optimal Linear Identification Mapping.
- Design of separable, analytically invertible feature extractor mappings for qualitative and quantitative image analysis.

Chapter 10

- Application of view-based pose estimation to multi-dimensional pose estimation.
- Design of pose tracking system.

Contents

Chapter 1

Introduction

This thesis brings together for the first time, the two disparate fields of synergetic pattern recognition and pose estimation. As a result of this combination, we have developed more powerful pattern recognition algorithms, more practical pattern learning algorithms and a fundamentally new approach to pose estimation.

In this chapter we introduce separately the fields of synergetic pattern recognition and pose estimation before combining them in the body of the dissertation.

1.1 What is Synergetics?

The word, synergetic, is derived from the Greek root *synergos*, which means cooperation. The term, synergetics, is used to convey two different meanings in this dissertation. This double meaning is historic, having been introduced by the man who coined the term. The two meanings, which are described below, are sufficiently different that it should not lead to any confusion. The intended meaning will be clarified explicitly in any instances where the context is not sufficient to do so.

1.1.1 Broad Meaning

The first meaning of synergetics then, is the broadest possible definition. Synergetics is the study of systems in which individual sub-systems act co-operatively. In particular, we are interested as to how this co-operation on a microscopic scale can lead to the formation of macroscopic spatial, temporal and functional structures.

Synergetic behaviour can be found in Bènard convection, a phenomenon from classical fluid dynamics. In this example, a liquid in a vessel is heated evenly from below, as shown in Figure 1.1. The heat is transported to the top of the vessel by conduction at a microscopic level, and no macroscopic motion or structure is visible. When the temperature difference between the top and bottom of the vessel reaches a critical temperature, however, macroscopic patterns, like those seen in Figure 1.2, may appear.

The individual molecules are now acting in a co-operative fashion to transfer the heat to the top of the vessel. The fluid forms visible rolls, rising at certain positions, cooling and then sinking down at different positions.

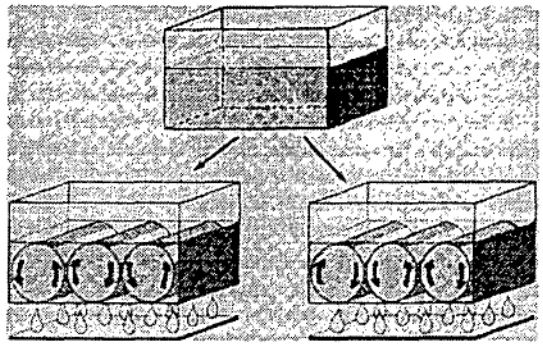


Figure 1.1: Onset of alternative roll patterns in a Bénard convection cell. From [27]

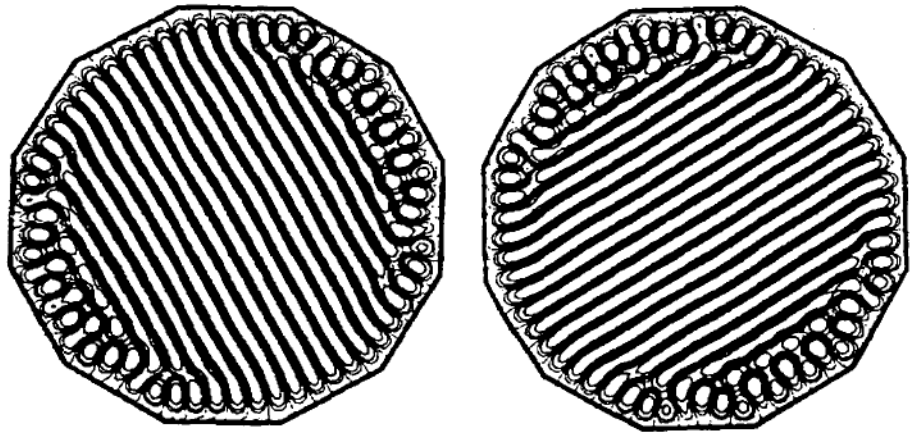


Figure 1.2: Two alternative possible roll patterns in a Bénard convection cell. From [27]

Referring again to Figure 1.2, it is clear that the shape of the vessel is symmetric, and the heating applied to the bottom surface is completely even. Yet this symmetry is broken by the rolls which adopt a specific orientation over all other possible orientations. Computer simulations confirm that the final orientation of the rolls is based on the initial conditions of the liquid, as can be seen in Figure 1.3. When, in the first two columns, the liquid is given an initial bias towards a given orientation, this modality grows until it dominates the system and all other modalities are suppressed. This can be seen when $t = 200$ in the bottom row. In the third column, the liquid was given biases towards two orientations. The two modalities grow and compete until the mode with the largest initial bias dominates the other. This *winner takes all* behaviour is typical of synergetic systems.

The breadth of this definition ensures that synergetics encompasses many disparate sciences. Among others, synergetic models have been used to describe chemical reactions [57], morphogenesis [11], neurobiology [4] and EEG-patterns [20, 26, 27] as well as spin-glasses and neural-networks [51, 106].

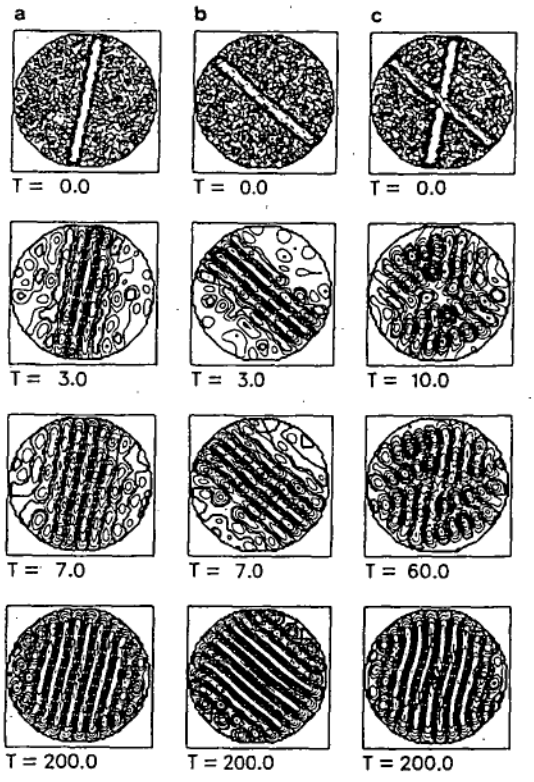


Figure 1.3: Compute simulations of a Bénard convection cell. Each simulation leads to a winner-take-all solution, dependent on the strengths of the competing modes at $t = 0$. From [8]

Synergetics is a category of systems, grouped together by similarities in their behaviour. It is not a group of techniques, but rather there is a large number of established techniques which are used to study synergetic systems. Among these are non-linear dynamics and bifurcation theory.

1.1.2 Specific Meaning

This dissertation investigates the application of synergetic systems to image analysis. We therefore restrict our attention to those systems in which synergetics produces macroscopic spatial structures, or patterns. In particular we study a canonical synergetic system, which was first introduced by Haken [38] as the basis of a competitive pattern recognition scheme. This system takes the form of a set of non-linear differential equations. It has a well defined set of user-controlled parameters, yet it is capable of exhibiting the behaviours typical of real-life synergetic systems. It is also amenable to the processing of image data due to its innate data reduction characteristics. These attributes make it an appropriate choice of synergetic system with which to attempt image analysis.

Much of the work in this dissertation is based on this particular system and extensions to it, as introduced by both ourselves and others. Our second, more specific meaning of the word synergetic, refers to systems, functions and algorithms based on this system.

1.2 What is Pose Estimation?

The *pose*, *orientation* or *attitude* of an object is the set of angles which define the rotation of the object around an arbitrarily defined set of axes. *Pose estimation* is the task of estimating the pose of a given object. *Pose tracking* is continuous pose estimation of an object over time.

In this dissertation we apply synergetic image analysis techniques to the challenge of pose estimation. While the physical object in question is three-dimensional, image analysis is clearly two-dimensional. Thus we face the challenge of estimating a vector of three-dimensions which describes the pose of a three-dimensional object using two-dimensional data.

1.3 Motivation

There are three main incentives to study synergetic image recognition with applications to pose estimation. First is the purely pragmatic assessment that these tasks have an extraodinary range of possible applications in automated industrial manufacturing environments. Second is the realisation that, while pose estimation is a very specific problem, casting it within the synergetic framework allows us to treat it as an instance of a very general problem with far-reaching implications. Finally, as synergetic pattern recognition is based on physical synergetic systems, we are greeted with the exciting prospect that algorithms created within this framework could be implemented on

massively parallel, analogue devices [37, 102]. We now look at each of these factors in turn.

1.3.1 Automating Industrial Manufacturing Environments

Image recognition systems already play a major, and increasingly important role, in modern manufacturing. Automated visual inspection (AVI) is a special case of pattern recognition, which is used to inspect, grade, measure and locate objects, often on an assembly line. The importance of AVI systems can be seen in the example of Motorola, which believes that almost without exception, each of its semi-conductor inspection tasks is 'a candidate for vision and automation' [18].

Pose estimation also has applications within manufacturing. A stationary robot, for instance, is generally required to manipulate an object on an assembly line. For this to be possible the robot must be able to recognise the object irrespective of its pose. It will also often need to estimate the pose to allow the robotic manipulator to grasp the object successfully.

A mobile robot faces these, plus other possible challenges with regard to pose. Assuming a robot needs to take a path that is blocked by objects of a known type, the robot's progress through the area would be enhanced by being able to estimate the pose, as well as the type of the object which was in the way, so as to be able to calculate the best possible path.

Appearance-Based Vision Systems

Traditional image analysis systems for manufacturing rely upon the existence of a CAD model of every object to be analysed. In essence, the standard approach involves extracting features from an image and comparing these with features extracted from the CAD model. While this approach has the ability to produce accurate and robust results, it is often slow and is expensive when the cost of creating a CAD model is high. In contrast, synergetic pattern recognition belongs to the expanding group of *appearance-based vision* systems [34], which can now offer a viable alternative to the traditional CAD model approach [85, 50, 63].

An appearance-based vision system is so-called because it constructs an implicit model of an object of interest through a systematic acquisition of views of the object [73, 15, 43]. This implicit model is not an attempt to recreate the CAD model, as might be done using *shape from shading* or *stereovision* routines. Rather, it is essentially a database which stores image features, indexed by the variables of interest to the user. The three major challenges facing researchers in appearance-based vision are how to design the features, how to construct the database and how to recall and interpolate between the test images. All three of these elements are addressed in this dissertation.

The characteristics of appearance-based vision systems are well suited to the nature of the manufacturing environment. First, training only requires a number of images of a given object. This makes it inexpensive and capable of being used generically for any object. Second, it is inexpensive to use, because the input to the system can be supplied by a single standard video signal. This is much cheaper than many alternative

sensing modalities, such as rangefinder, ultrasound or x-ray imaging. Third, as well as being inexpensive, this sensing modality means that the process is completely *passive*, so it does not interfere with the manufacturing process.

As implied by the name, appearance-based vision systems rely on the appearance of an object being consistent, so they are sensitive to changes in external variables such as lighting conditions. Fortunately, the highly controlled nature of many manufacturing environments means that the lighting conditions, and the camera parameters can be fixed.

1.3.2 Working Within a General Framework

Image recognition and pose estimation are two instances of a broad field of study known as *machine vision* which attempts to determine values of interest to the user from an image. This in turn is a subset of *pattern analysis*, which attempts to do the same, this time using any type of input.

Synergetic image recognition and pose estimation as espoused in this work are generic in two senses. First, they are generic in that the features required by the system are *object independent*. In contrast to systems that have been designed to work on a specific object, this makes the algorithms flexible enough to work on any object.

Beyond this, however, it is generic in that the variables that are estimated by the system can be freely chosen by the user. For example such a system could be trained to estimate the location of the predominant light source, instead of object pose. This flexibility means that our algorithms can be used as general purpose *machine vision* algorithms.

Furthermore, our treatment of the input images is not, with the exception of the method described in Chapter 7, specific to images. For example, we do not extract lines or locations of vertices and other visual highlights. Instead the pixels are simply treated as a vector of inputs in a generic way. Thus, we can see that the algorithms can be thought of as general purpose *pattern analysis* algorithms.

By following the philosophy of maintaining a generic approach, one loses a number of opportunities to increase accuracy by using problem-specific knowledge. The payback, however, is that advances made in the specific case can be directly applied to the entire range of problems within *machine vision* and *pattern analysis*.

1.3.3 Producing Synergetic Devices

The dominant computing paradigm of the Turing machine [97] is the basis of the modern digital computer. While these devices are capable of massive numerical calculations that the human mind could not contemplate, they are very poor at dealing with problems of classification, rejection and generalisation which even a child can solve successfully.

Artificial neural networks (ANNs) are an alternative approach to computing, inspired by the structure of the human brain. Researchers have demonstrated that even small ANNs can classify, reject and generalise as well as learn and forget. While most ANN research is implemented in software on a standard digital computer platform,

there is no doubt that much of the excitement about ANNs stems from the prospect that a viable hardware implementation will be possible in the future. Unfortunately, the massive connectivity required of an ANN chip would make such a device incredibly expensive to produce.

In a similar way, synergetic pattern recognition represents an alternative approach to computing, inspired by physical synergetic systems. Also similarly, the goals of synergetic pattern recognition are to classify, reject and generalise. While our implementation is in software for a digital computer, the prospect of a viable hardware implementation seems a lot more likely than for ANNs [37, 102]. This is because the systems which we are modelling are much simpler than the human brain. They also occur commonly around us.

1.4 Challenges and Achievements

This dissertation tackles several of the key issues facing synergetic pattern recognition and pose estimation.

Generalisation

The study of synergetic pattern recognition is still in its infancy. As such, there are many important issues that need to be addressed. Foremost among these issues is the ability of such a system to *generalise* from the set of patterns on which it has been trained. The current state-of-the-art in synergetic pattern recognition requires that we accept a compromise between the power of that generalisation and the speed of the classification. This is a serious limitation to the practical situations in which synergetic pattern recognition can be used and represents a major disadvantage in comparison to standard pattern recognition approaches.

We address this issue by introducing two new synergetic algorithms which increase the *classification power* of the system. The algorithms are extensions of the standard approach in which certain constants are allowed to vary, thereby parameterising the location of the class boundaries. We also introduce corresponding learning techniques which minimise the classification error over a given training set.

Rejection

When considering pattern recognition schemes we must be aware not only of the challenge of classifying patterns correctly, but also of rejecting patterns that do not belong to any of the classes. The model used to define synergetic pattern recognition does not allow for this concept, and no serious attempt has been made to address this in the literature. The ability to reject is of particular importance when considering synergetic devices in hardware, because we do not want noise internal to the device being classified as a pattern.

It is possible to set arbitrary rejection levels and enforce these restrictions directly, but such an approach breaks the link between the recognition algorithm and the canonical synergetic model. So we follow a more sophisticated approach and extend the synergetic model to include a rejection well. The well is parameterised by a number of

variables which allow the user to simply train the rejection system appropriately for a given training set.

Learning

In an even earlier stage of development is the idea of learning using synergetic algorithms. The advantage of synergetic learning is that it occurs using the same process as synergetic recognition, which as well as being elegant, is in accord with neurophysiological evidence that learning and recall are parallel processes. Unfortunately, the current approach to synergetic learning involves a system of very high dimensionality, and requires the addition of extra terms to the learning model. As a result, the strong relationship between learning and recognition is significantly weakened and the resulting system quickly becomes impractical for even moderate numbers of classes. The system is also no longer based on the minimisation of a Lyapunov function and so the behaviour of the system is unpredictable.

We remove these issues by recognising a fundamental relationship between the training images and the learned memories. This understanding allows us to construct a new learning algorithm. The new technique maintains the strong link between synergetic recognition and synergetic learning, is based on the minimisation of a Lyapunov function and is of a much lower dimension than the current synergetic learning algorithm.

Continuous Variable Estimation

The challenge of applying synergetic pattern recognition to pose estimation stems from the fact that pattern recognition is a *discrete* task, whereas in pose estimation the results are *continuous* variables.

This is the first work which has attempted to develop a continuous form of synergetic pattern recognition. In fact, we have developed three distinct approaches to this problem, each of which is described in a separate chapter.

Estimation Speed

While much work on the accuracy and robustness of pose estimation systems has appeared in the literature, the third element which determines the applicability of an approach to pose estimation is speed. This is particularly the case when attempting to track the pose of an object in real time. For such a task, the time requirements are paramount in determining the success of the solution. The fundamental approach to this problem is consistent throughout the literature and essentially involves searching a database of points in a high-dimensional space. A number of clever search variations exist to help speed the process, but the search is still the bottleneck in the estimation system and the main impediment to real-time pose estimation.

Our solution to this problem is the most exciting finding of this work. It is a fundamentally different approach which completely removes the need for a search through a series of database points. We call it *explicit inversion*, because the system is designed such that we invert a known function explicitly, rather than an unknown function implicitly. We show that explicit inversion yields accuracy comparable to

current algorithms but with a dramatic reduction in estimation times. We have found reductions of between one and two orders of magnitude. Our technique has a number of other powerful properties, including the ability to de-couple independent parameters and the existence of an optimum training strategy. Furthermore, explicit inversion has possible applications wherever pattern analysis is required, thus justifying our generalist approach to synergetic pattern recognition and pose estimation.

1.5 Dissertation Structure

The structure of the dissertation has been built around meeting the challenges just identified. At the highest level, it has two parts. Part A describes synergetic pattern recognition and learning.

Chapter 2 introduces the concept of synergetics in physical systems and shows how we can model such systems to derive synergetic pattern recognition algorithms. We review the current range of synergetic pattern recognition systems and recognise their major strengths and weaknesses.

Chapter 3 addresses the fact that the current crop of synergetic pattern recognition algorithms are highly constrained in their classification power. We introduce two new pattern recognition algorithms based on a generalisation of the canonical synergetic model. Both of these algorithms can be optimised over a training set to offer improved classification power.

Chapter 4 tackles the fact that the standard synergetic pattern recognition system is incapable of rejecting an image. We introduce an extension of the standard synergetic model which incorporates a parameterised, trainable rejection well. This new model is then used as the basis of a synergetic pattern recognition and rejection system.

Chapter 5 investigates the concept of *unsupervised learning* in synergetic pattern recognition. We review the current approach to this problem and find it to be unwieldy and incapable of reproducing the results of supervised learning. We propose a new approach to unsupervised synergetic learning which addresses both of these issues.

Part B discusses how to apply synergetic pattern recognition to the task of pose estimation.

Chapter 6 reviews the concept and specific instances of view-based pose estimation from the literature. It also introduces the key ideas of uniqueness, equality and ambiguity.

Chapter 7 then introduces a new approach to pose estimation based on the results of synergetic pattern recognition and neuro-physiological theories of the human visual understanding of pose.

Chapter 8 describes a second, more practical approach to view-based synergetic pose estimation and compares results with the benchmark routine in this field.

Chapter 9 introduces the new technique of *explicit inversion*. Results on the database of images used in Chapter 8 show that this new approach yields comparable accuracy to current algorithms but with a dramatic reduction in calculation times. For the particular data set in question, we found reductions of one order of magnitude.

Chapter 10 reports on the application of explicit inversion to estimating the pose of an aircraft. This work was carried out in response to a problem faced by Australia's Defence Science and Technology Organisation. We show that the increase in speed offered by the explicit inversion technique makes a real-time pose tracking system possible.

Chapter 11 draws conclusions from the work, places it in a more general context and proposes future directions for research.

1.6 Notes on Notation

The following general rules have been used with notation throughout this dissertation:

q	scalar
$\boldsymbol{q}$	vector
Q	matrix
$\dot{q}$	rate of change of q with respect to time
$\dot{\boldsymbol{q}}$	rate of change of $\boldsymbol{q}$ with respect to time
$\dot{Q}$	rate of change of Q with respect to time
q_i	the ith of a series of q's or the ith element of $\boldsymbol{q}$
$\boldsymbol{q}_i$	the ith of a series of $\boldsymbol{q}$'s or the ith vector of Q
Q_{ij}	the element of Q from the ith row and jth column
R_x	rotation around the x axis

Part A

Synergetic Pattern Recognition and Learning

Chapter 2

Synergetics and Pattern Recognition

Synergetic systems are comprised of individual microscopic sub-systems which act in a co-operative way to from macroscopic structures. As can be seen in the Bénard convection cell of Figure 1.2, we can consider a synergetic system as a *pattern formation* system.

In order to use such systems to create a synergetic pattern recognition scheme, we need to achieve three goals. First, we need a better understanding of the natural systems. Second, we need the ability to create, using the same principles, artificial pattern formation systems which form particular, user-given patterns. Third, we must understand how to use such a system for the task of pattern recognition. These three goals are tackled in Section 2.1.

Having retraced the derivation of the standard synergetic approach to pattern recognition, we then review the variations which have been proposed in the literature and analyse their major strengths and weaknesses.

2.1 Deriving a Synergetic Pattern Recognition System

2.1.1 Defining the Synergetic Pattern Formation Model

In order to better understand how pattern formation in natural synergetic systems works, we need a model which is capable of reproducing the characteristics of a synergetic system while requiring only a manageable number of system parameters. As it is capable of producing synergetic behaviour, we will label this a synergetic model.

The exposition below closely follows the initial formulation of synergetic pattern recognition by Haken [38], while changing the notation to be consistent with the rest of the dissertation.

We start by summarising the results of this section, such that the important concepts are clear within the mathematical details. When a system is controlled externally, the system may be driven away from the equilibrium of a stable state into an unstable state. The point at which this occurs is a phase transition. When a phase transition occurs, there are a number of possible modes or patterns that the system could ex-

hibit. The various modes compete against each other until one mode dominates and damps all the other modes. The amplitudes of the growing modes are called *order parameters*. Which mode dominates the system is dependent on the initial state of the system, as produced by fluctuations.

The analysis begins by defining a highly general set of differential equations which control the state of the synergetic system. Based on experience with physical synergetic systems, such as the Bénard convection cell, we assume that there is a stable state and proceed with a linear stability analysis. Having established conditions for the stability of the state, we construct a form for the state vector as a function of both space and time. This form allows us to re-express the initial set of differential equations in terms of a set of differential equations for the order parameters. The eigenvalues associated with each system mode are then used to invoke Haken's *slaving principle*, by which it is found that the majority of the system's modes are not independent, but rather are slaves to a small number of dominant modes. The dynamical system has now been reduced to one equation of motion for each dominant mode, which makes clear the possible behaviours of the system. By re-expressing the motion as the movement of a particle on a potential surface, we can isolate each of the important macroscopic behaviours.

The state of a physical system can be defined by a state vector $\boldsymbol{q}$ of length l. As we require that our model be capable of producing spatio-temporal patterns, each element of the vector $\boldsymbol{q}$ should be dependent on both space and time,

$$\boldsymbol{q} = \boldsymbol{q}(x, y, z, t) = \boldsymbol{q}(\boldsymbol{x}, t). \tag{2.1}$$

Each element of the state vector $\boldsymbol{q}$ describes a physical measure of the system. In a fluid, for example, you may have three elements; the density $\rho(\boldsymbol{x}, t)$, the velocity $\boldsymbol{v}(\boldsymbol{x}, t)$ and the temperature $\tau(\boldsymbol{x}, t)$. In a chemical reaction, $q_i(\boldsymbol{x}, t)$ might represent the concentration of the ith chemical in the system.

The dynamics of the system are modelled by a set of differential equations of the form,

$$\dot{\boldsymbol{q}}(\boldsymbol{x}, t) = \boldsymbol{d}[\boldsymbol{q}(\boldsymbol{x}, t), \nabla, \boldsymbol{\alpha}, \boldsymbol{x}] + \boldsymbol{f}(t). \tag{2.2}$$

The vector function $\boldsymbol{d}$ defines the deterministic evolution of the state vector $\boldsymbol{q}$, while $\boldsymbol{f}(t)$ describes fluctuating forces which may be either internal or external to the system. We now look briefly at each of the dependencies in the deterministic function $\boldsymbol{d}$. Clearly the rate of change is dependent on the current state of the system, $\boldsymbol{q}(t)$, possibly over the entire space domain of $\boldsymbol{x}$. The nabla operator, ∇, allows the inclusion of diffusion or wave propagation within the system by incorporating the partial differentials, $\nabla = (\partial/\partial x, \partial/\partial y, \partial/\partial z)$. The vector $\boldsymbol{\alpha}$ represents external controls on the system. In the Bénard convection described in Chapter 1, for example, the temperature difference between the top and bottom plates is the single external control. In the language of bifurcation theory, the external controls are called bifurcation parameters. Finally, $\boldsymbol{d}$ may also be dependent on spatial inhomogeneities, as indicated by the inclusion of $\boldsymbol{x}$ in the dependencies of $\boldsymbol{d}$.

Equation 2.2 is a very general prescription, capable of a wide range of behaviours, and applicable across distinct fields, such as biology, physics, chemistry and fluid

dynamics. It is also impossible to solve for the general case. Based on a knowledge of bifurcation theory, changes in the bifurcation parameters or external controls can lead to qualitative changes in the state of such systems, so to proceed, we assume that there is a time independent, stable state $\boldsymbol{q}_0$,

$$(\boldsymbol{\alpha} = \boldsymbol{\alpha}_0) \Rightarrow (\boldsymbol{q} = \boldsymbol{q}_0), \tag{2.3}$$

and carry out a linear stability analysis.

We perturb the system in its steady state by changing the external control,

$$(\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \delta) \Rightarrow (\boldsymbol{q} = \boldsymbol{q}_0 + \boldsymbol{w}(\boldsymbol{x}, t)). \tag{2.4}$$

Substituting this form of solution into Equation 2.2, expanding the nonlinear function $\boldsymbol{d}$, and temporarily ignoring the fluctuation terms, we find

$$\boldsymbol{d}(\boldsymbol{q}_0 + \boldsymbol{w}) = \boldsymbol{d}(\boldsymbol{q}_0) + L\boldsymbol{w} + \hat{\boldsymbol{d}}(\boldsymbol{w}), \tag{2.5}$$

where L is the linearisation matrix, defined by,

$$L_{ij} = \frac{\partial d_i}{\partial q_j} \quad \text{at } \boldsymbol{q} = \boldsymbol{q}_0, \tag{2.6}$$

and $\hat{\boldsymbol{d}}(\boldsymbol{w})$ contains the higher order terms. By assumption, $\boldsymbol{q}_0$ is a stationary point and $\boldsymbol{w}$ is small, so the first term on the right hand side of Equation 2.5 is zero and the third term is small. Thus the linear stability analysis yields the linear differential equation,

$$\dot{\boldsymbol{w}} = L\boldsymbol{w}. \tag{2.7}$$

For the case of non-degenerate eigenvalues then, we can find solutions for $\boldsymbol{w}$ in terms of the eigenvalues and eigenvectors,

$$\boldsymbol{w} = e^{\lambda_j t}\boldsymbol{v}(\boldsymbol{x}_j), \tag{2.8}$$

and substitute back into Equation 2.4 to obtain,

$$\boldsymbol{q} = \boldsymbol{q}_0 + \sum_j \xi_j(t)\boldsymbol{v}_j(\boldsymbol{x}), \tag{2.9}$$

where $\xi_j(t)$ are the *order parameters*. Now clearly in Equation 2.9 the magnitude of the order parameter measures the strength of the associated eigenvector. Thus by inspecting the values of the order parameters, we can investigate the status of the competition between modes. We can do this by substituting the new form for $\boldsymbol{q}$ into the original general formulation of Equation 2.2, thereby re-expressing the system as a set of differential equations for the order parameters.

$$\sum_j \dot{\xi}_j(t)\boldsymbol{v}_j(\boldsymbol{x}) = \boldsymbol{d}[\boldsymbol{q}_0 + \sum_j \xi_j(t)\boldsymbol{v}_j(\boldsymbol{x}), \nabla, \boldsymbol{\alpha}, \boldsymbol{x}] + \boldsymbol{f}(t). \tag{2.10}$$

To simplify the resulting expression, we apply the linearisation of Equation 2.5, and introduce the adjoint eigenvectors, $\boldsymbol{v}_k^+(\boldsymbol{x})$ which are constructed to be orthonormal to the eigenvectors $\boldsymbol{v}_k(\boldsymbol{x})$,

$$\boldsymbol{v}_i^+ \boldsymbol{v}_j = \delta_{ij}, \tag{2.11}$$

where δ_{ij} is the Kronecker delta function. We multiply Equation 2.10 by $\boldsymbol{v}_k^+(\boldsymbol{x})$ and integrate over all space to find,

$$\dot{\xi}_k = \lambda_k \xi_k + \hat{d}_k(\boldsymbol{\xi}) + \hat{f}_k(t), \tag{2.12}$$

where $\hat{d}_k(\boldsymbol{\xi})$ contains nonlinear deterministic components dependent on the entire vector of order parameters and $f_k(t)$ represents the fluctuating forces acting on the kth order parameter over time.

It is clear from Equation 2.8 that modes with a positive eigenvalue will grow in amplitude with time. We categorise these modes as *unstable*. Equally, modes with negative eigenvalues will decay with time and are labelled *stable*. The case of complex eigenvalues leads to time-periodic solutions which are not required for an understanding of this dissertation. So restricting ourselves to real eigenvalues, we can re-write Equation 2.12 to represent the two mode classes,

$$\begin{aligned} \dot{\xi}_u &= \lambda_k \xi_u + \hat{d}_u(\boldsymbol{\xi}) + \hat{f}_u(t), \\ \dot{\xi}_s &= \lambda_k \xi_s + \hat{d}_s(\boldsymbol{\xi}) + \hat{f}_s(t), \end{aligned} \tag{2.13}$$

where the subscripts u and s designate unstable and stable respectively. Now we invoke Haken's *slaving principle* [39], which states that when the real parts of λ_u are small, the stable order parameters are driven by the unstable order parameters, and can therefore be written as,

$$\xi_s = g_s(\xi_u(t), t). \tag{2.14}$$

We can therefore rewrite Equation 2.12 without explicit reference to the stable modes as,

$$\dot{\xi}_u = \lambda_k \xi_u + \tilde{d}_k(\boldsymbol{\xi}) + \hat{f}_u(t), \tag{2.15}$$

where the competition between unstable modes for dominance of the system is implicit in the nonlinear term, $\tilde{d}_k(\boldsymbol{\xi})$.

Now we wish to make this formulation more concrete by describing the type of intermodal competition. We assume that the competition is analogous to the evolution of a critically damped particle moving on a potential surface, $p(\boldsymbol{\xi})$, giving us a special case of Equation 2.15 where

$$\dot{\xi}_u = -\frac{\partial p}{\partial \xi_u}. \tag{2.16}$$

This now gives us a firm intuitive feeling for the evolution of the system. A particle will move along the lines of steepest descent until it lands in a well on the potential surface. This well represents one of the many possible final outcomes for the system. The starting point of the system, combined with the potential surface, determines the final state of the system.

2.1.2 Constructing an Artificial Synergetic Pattern Formation System

Having gained a better understanding of pattern formation processes in natural systems, our next step is to adapt our model to form specific, user defined patterns.

We start by expressing the concept of a pattern as a state vector. Restricting ourselves to greyscale images, the standard treatment of an image by digital computers is to digitise the continuous, two dimensional luminance function into a matrix of pixel values, each of which is proportional to the average luminance over the pixel. We can then simply reshape the matrix into a column vector of n pixel values to represent the state vector $\boldsymbol{q}$ of the system.

Now we need to construct a suitable $\boldsymbol{q}_0$, which is an unstable, time independent stationary state. The most obvious choice is the vector containing all zeros which represents the completely blank pattern. We can then re-express the state vector as

$$\boldsymbol{q} = \sum_u \xi_u(t)\boldsymbol{v}_u(x) + \boldsymbol{w}_s, \tag{2.17}$$

where $\boldsymbol{w}_s$ represents the effect of the stable modes, which tends to zero as $t \to \infty$.

In the previous section we derived the possible final states from the linearisation of an unknown function. Now we proceed in the opposite direction and directly define the m memories, $\boldsymbol{v}_k$, which we call *prototypes* in the language of pattern recognition. For the sake of neatness, we scale the prototype vectors to have unit length. In order to construct a set of *adjoint prototypes*, $\boldsymbol{v}_k^+$, the prototypes also need to be linearly independent. This requirement is almost automatically fulfilled by the nature of the inputs. The number of pixels in each image, n, will invariably be much larger than the number of prototypes, m, so the likelihood that the patterns will be linearly dependent is very small. The matrix of adjoint prototypes which satisfies Equation (2.11) can then be given as,

$$V^+ = (V^T V)^{-1} V^T, \tag{2.18}$$

where V is the $n \times m$ matrix containing the prototype column vectors and the superscript T is the matrix transpose operator. The kth row in V^+ is the kth adjoint prototype, $\boldsymbol{v}_k^+$.

2.1.3 A Pattern Recognition System Based on the Synergetic Model

The final step in deriving Haken's synergetic pattern formation system is to introduce the important concept that we can implement *pattern recognition* as an instance of pattern formation. In fact, Haken [40] goes further and states his belief that pattern recognition *is* pattern formation.

In this section we show how to construct a potential function, p, such that the only possible final states of the system are the user defined prototypes. Now if an image is given to the system as initial conditions, the dynamics on the potential surface will evolve such that the new image is transformed into one of the prototypes. We consider this process to be pattern recognition, and so the new image is classified as belonging to the same class as the selected prototype.

High Dimensional System

We now introduce a potential function p which leads to a dynamical system analogous to Equation 2.15. When choosing a functional form for the potential, we have a number of requirements. First, we need the potential to be differentiable so that we can derive a dynamics that minimises the value of the potential. Second, we need the potential to have minima for each prototype image, and no other spurious minima.

Haken introduced the following potential form which satisfies these criteria,

$$p = -\frac{1}{2}\sum_{k=1}^{n}\lambda_k(v_k^+ q) + \frac{1}{4}\sum_{l\neq k}\sum_{k\neq l}B_{kl}(v_l^+ q)^2(v_k^+ q)^2 + \frac{1}{4}c(q^T q)^2, \tag{2.19}$$

where B_{kl} and c are constants and $\lambda = [\lambda_1, \ldots, \lambda_n]$ is a vector of constants labelled *attention parameters.*

In fact, this formulation is symmetric about the origin such that the potential is minimised when $q = \pm v_k$, $\forall k$. So the framework of synergetic pattern recognition considers inverted images to be identical.

Each of the three terms on the right hand side of Equation (2.19) plays a distinct role in defining the potential surface. The first term defines the minima on the potential surface at the prototypes. The depth of each minimum is controlled by the attention parameters, λ_k. The second term defines the competition among prototypes and controls, in combination with the attention parameters, the location of the ridges in the potential surface. This is parameterised by the matrix constant, B_{kl}. The third term is required to limit the growth of q, and is parameterised by the constant c.

Applying the gradient descent approach of Equation 2.16 yields,

$$\dot{q} = \sum_{k=1}^{n}\lambda_k v_k(v_k^+ q) - \sum_{l\neq k}\sum_{k\neq l}B_{kl}(v_l^+ q)^2(v_k^+ q)v_k - c(q^T q)q. \tag{2.20}$$

Now to use this dynamical system for pattern recognition, we select a set of prototypes, calculate the adjoint prototypes using Equation 2.18 and use a numerical integrator to track the evolution of a novel image into one of the stored memories.

Low Dimensional System

The dynamic system which results from the potential of Equation (2.19) has one dimension for each pixel in the test image, which is a massive computational burden. In order to make the process viable on a digital computer, we use the fact that the evolution of the system is controlled completely by the *order parameters*, ξ_k. We calculate the initial values for the order parameters by projecting the image onto the adjoint prototypes,

$$\xi_k = v_k^+ q, \qquad k = 1, \ldots, n. \tag{2.21}$$

Using this definition and applying the orthonormality conditions of Equation 2.11, it follows that the kth prototype is projected onto the kth axis unit vector in order-parameter space.

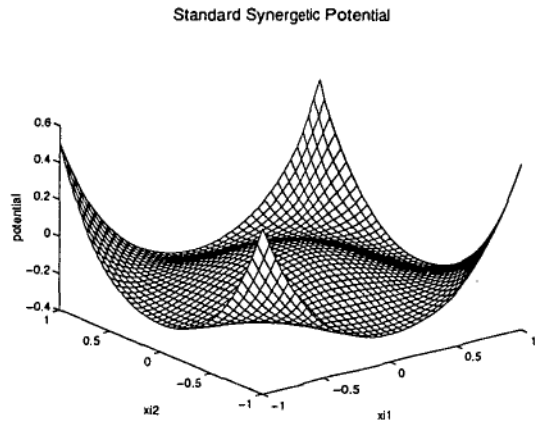


Figure 2.1: The restricted synergetic potential with $\lambda = 1$.

We can now re-write the potential as a function of the n order parameters,

$$p = -\frac{1}{2}\sum_{k=1}^{n}\lambda_k\xi_k^2 + \frac{1}{4}\sum_{l\neq k}\sum_{k\neq l}B_{kl}\xi_l^2\xi_k^2 + \frac{1}{4}c\left(\sum_{l=1}^{n}\xi_l^2\right)^2. \tag{2.22}$$

Finally, we now use Equation 2.16 to derive a low dimensional dynamic system,

$$\dot{\xi}_k = \lambda_k\xi_k - \sum_{l\neq k}B_{kl}\xi_l^2\xi_k - c\left(\sum_{l=1}^{n}\xi_l^2\right)\xi_k. \tag{2.23}$$

This is Haken's pattern formation model, which we label PF and use as the basis of synergetic pattern recognition.

2.2 Deriving a Linear Synergetic Pattern Recognition Algorithm

2.2.1 Predicting the Final State

In order to gain an intuitive understanding of synergetic pattern recognition, we now look at the potential surface defined by Equation 2.22. In order to visualise the surface, we fix all but one of the $n^2 + n + 1$ free parameters, so that,

$$(B_{kl} = c = \lambda_k) = \lambda > 0, \tag{2.24}$$

and plot the resulting surface in Figure 2.1 for the case $\lambda = 1$.

The pattern formation model resulting from this choice of parameters is labelled PF_R, where the subscript R denotes, 'restricted':

$$\dot{\xi}_k = f_k(\boldsymbol{\xi}, \lambda) = \lambda\xi_k(1 + \xi_k^2 - 2\sum_{l=1}^{n}\xi_l^2). \tag{2.25}$$

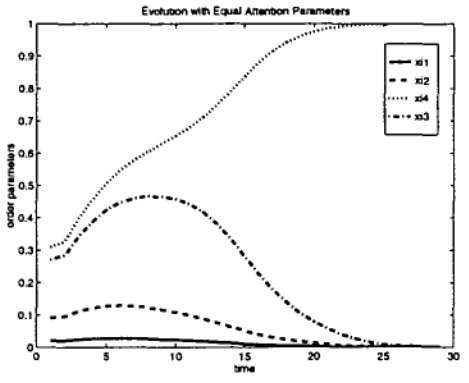


Figure 2.2: The final state of the order parameters determines the classification.

We can clearly see that this potential surface satisfies our needs as the basis for an associative memory. Furthermore, the symmetry of the surface suggests that it may be possible to predict the final position of the imaginary particle from its initial conditions. Indeed, Haken [40] has proven that PF_R has the following properties:

- all the prototypes (and their negatives) are possible final states;
- there are no other possible final states;
- the order parameter with the largest absolute value will grow while all others will decay.

The first result assures us that all memories are accessible. The second result proves that the system will not learn any 'spurious' memories, as is often the case in neural network associative memories. The third property implies that the final solution can be predicted from the initial conditions, so that integrating the system is unnecessary for classification. Figure 2.2 demonstrates behaviours typical of the PF_R model. The final state of the system is $\xi_4 = 1$ and $\xi_i = 0$, $\forall i \neq 4$. This signifies a classification of class 4. Note that the classification could have been predicted by inspecting the initial order-parameter set.

2.2.2 SCAP

These properties of the PF_R model led to the creation of SCAP, which stands for Synergetic Computer using Adjoint Prototypes. SCAP is a linear algorithm which calculates the initial order-parameter set using Equation (2.21) and classifies the image according to the order parameter with the largest absolute value. The SCAP algorithm has been used successfully in industrial settings [101], due mainly to its high speed classification.

2.3 Other Synergetic Pattern Recognition Algorithms

SCAP is the most widely used synergetic pattern recognition system, but it is far from the only scheme which can be drawn from the synergetic framework described above.

Here we describe a number of alternatives in the literature, divided into four categories based on the way in which they differ from SCAP. The first category uses the same dynamics but uses a more sophisticated way of constructing the prototype images. The second category removes some of the restrictions on the pattern formation variables used in PF_R. The third category includes the diffusion term, ∇^2, which was excluded when deriving SCAP. The fourth category extends the standard approach by including pre-processing for synergetic pattern recognition systems.

2.3.1 Choosing Prototypes

The approach to pattern recognition derived from synergetic pattern formation is example-based. The standard approach requires that the user select one example for each class which is representative of that class. Clearly, this is not a simple task, as it is rare that the variations within a class of patterns can be sufficiently represented by a single prototype. Assuming that the user has access to a number of images from each class, three schemes have been proposed in the literature for the creation of *hybrid* images to be used as prototypes.

Average Image

The first obvious approach is to use prototypes which are the average image for each class in the training images. This idea has the significant drawback that if a training image were given to the system it will not be projected directly onto one of the minima. In fact, we cannot be certain that even the training set will be classified correctly.

SCAPAL

SCAPAL [101] is an iterative training method designed to construct the hybrid prototype which minimises the classification error. It does this by focusing the training of the system on those training images which are important for the learning process, as suggested by the experimental findings of Anderson and Gaborsky [1], who found that repeatedly training neural network systems on mis-classified images can substantially reduce the error rate. The starting point for the iteration are the averaged prototypes described above, and the iterations are stopped when the classification error reaches a specified level or no longer decreases. The process is described in point form below.

- **construct** a SCAP classifier using the current averaged prototypes v_k.
- **calculate** the average δv_k over the mis-classified training images.
- **adjust** the average prototypes by setting $v_k = v_k + \alpha \delta v_k$, where α is an adjustable learning rate.
- **return** to first step.

MELT

A practical solution to the problem of allowing multiple prototypes per class was proposed by Böbel et al. [12]. In their approach the orthonormality between the prototypes and adjoint prototypes is generalised such that,

$$\boldsymbol{v}_i^+ \boldsymbol{v}_{jl} = \delta_{ij}, \tag{2.26}$$

where $\boldsymbol{v}_{jl}$ is the lth training image belonging to class j. We are therefore looking for a set of adjoint prototypes such that each training image belonging to class j is projected onto the jth unit axis in order parameter space.

We calculate such a set of adjoint prototypes by defining $\hat{I}$ as a generalised unit matrix. $\hat{I}$ has one column for each training image, one row for each class and each column has a single element equal to unity with all other elements being set to zero. For example, the $\hat{I}$ given below,

$$\hat{I} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \tag{2.27}$$

is for the situation where the first three prototypes belong to class 1, the next two prototypes to class 2, and the final two prototypes are from class 3.

Now the required adjoint prototypes can be written in the same form as Equation (2.18),

$$V^+ = \hat{I}(V^T V)^{-1} V^T. \tag{2.28}$$

This process has essentially reduced the multiple training images for a given class to form a single prototype image by 'melting' them together. This is the basis of the name MELT that has been given to this method. In this way, the multiple training image per class problem has been reduced to the original problem and we can therefore proceed using the high-dimensional dynamics, the low-dimensional dynamics, or the linear classifier used for SCAP.

2.3.2 Removing Parameter Restrictions

Selective Competition

The pattern formation model PF of Equation 2.23, includes a symmetric matrix B of *competition parameters*. By adjusting these parameters, we can set a different level of competition between each pair of prototypes. Yudashkin [108] has used this ability to allow multiple prototypes per class. He constructs a B such that,

$$B_{kl} = \begin{cases} b & k \text{ and } l \text{ different classes} \\ 0 & k \text{ and } l \text{ same class.} \end{cases} \tag{2.29}$$

Now using this competition matrix along with the restrictions,

$$(c = \lambda_k) = b > 0, \tag{2.30}$$

we have a new pattern formation model, $PF_{Yudashkin}$.

The competition between classes means that the prototypes of all the non-winning classes will be suppressed and evolve to have zero order parameters. Because there is no competition between prototypes of the same class, however, the final solutions for the system will no longer be the unit axes, as was the case with PF_R. Instead, there will exist a curved manifold between the unit axes in order parameter space which is stable and which represents the selection of a winning class.

Attention Parameters

The attention parameters, λ, control the depth of each minimum on the synergetic potential surface and are, therefore, key in controlling the evolution of a particle moving on the surface. For example, if the attention parameter for the ith class is set such that their are no minima for that class, then no image will be classified as belonging to the ith class. Haken [40], has used this control to model the recognition of several different objects within one scene, and our oscillating perception of ambiguous patterns.

Wang et al. [103] loosened the restrictions on the attention parameters in PF_R to generalise the pattern formation model, and hence the pattern recognition system. They found that by selecting a set of attention parameters which minimise the classification error over a training set of images, they could correctly classify images that were incorrectly classified using SCAP. Unfortunately they did not find a method of predicting the outcome of the evolution so, although it is a more powerful classifier, their system is much slower than SCAP, and their scheme for training the attention parameters is not guaranteed to take advantage of this extra classification power. More details of this work are given in Chapter 3, where we extend this concept and produce a classifier that is as powerful as this approach, yet has the same speed as SCAP.

2.3.3 Synergetics with Diffusion

The standard synergetic pattern recognition model, and indeed, most associative memories [59, 2, 67, 28], treat two dimensional images as one dimensional vectors. Yet it is well known that neighbouring pixels have a higher correlation than distant pixels, so spatial information is lost in this representation.

It is also likely that the human visual system processes information in parallel, using local calculations only [56]. In contrast the models mentioned above work sequentially and use information from the entire image to create the dot products.

These facts are the motivation behind two extensions to the standard synergetic model which introduce local, parallel processing by adding diffusive terms to the standard system.

The first of these extensions was proposed by Schmutz and Banzhaf [91], who added standard nearest-neighbour diffusion between neurons and identified the existence of localised stable states when a balance between the diffusive and localising effects of the system has been reached. The existence of these diffuse stable states in the model matches well with known neurobiological systems which exhibit coherent spatial structures over lattices of neurons.

As well as being representative of the states found in neurobiological systems, these ground states are much more robust than those found in the standard model, because the information is stored across the lattice instead of in a single neuron. In the original formulation damage to the neuron which recalled class i would result in the system failing to recognise any instance of class i. Bressloff [13] has proven that these states can exist in any finite-dimensional lattice.

The second of these extensions adds diffusion to the original system, but structurally separates the reaction and diffusion effects. Yuasa et al [107] have extended the concept of the order parameter from a scalar value for each prototype, to a two dimensional matrix for each prototype. Each matrix element represents the *activity* of an individual pixel, and changes in time as the system evolves. They introduce diffusion with periodic boundary conditions that acts not on the pixel values, but on the activity matrix. The diffusion acts to average out the activity levels *within* each prototype. At the same time, the standard synergetic competition is taking place *between* the prototypes. The combination of these two effects means that the system has final states where the activity matrix of one prototype is uniformly equal to unity, while all others are equal to zero.

The extended system is therefore capable of emulating the results of the original system, but the local processing decreases classification time, and greatly reduces the number of connections required in a possible hardware implementation.

2.3.4 Synergetics with Pre-processing

Haken and his colleagues have developed a number of pre-processing techniques, to be used in conjunction with standard synergetic recognition, which increase the system's robustness to important types of transformations.

The first approach is a *static* pre-processing in which each image is transformed in such a way that the resulting feature set is identical for a given image irrespective of how it has been transformed. For translation, scale and in-plane rotation, Haken [40] has proposed using Fourier transforms and logarithmic maps to achieve this invariance. If all of the training and test images are pre-processed identically, it is clear that the standard pattern recognition system will be invariant to these transforms.

The second approach is a *dynamic* pre-processing. This procedure can be used for scaling, translation and in-plane rotation [40], as well as for more complicated transformations, such as small, local deformations across an image [40, 16, 17]. There are a number of pattern recognition problems in which we would like to introduce invariance to small deformations, such as in optical character recognition.

Using this idea, the particular transformation is parameterised so that the test image can be transformed into any one of a family of images. The transformation changes the pixel values indirectly by changing the location of points on the image. So if $\boldsymbol{x}$ defines a square grid of pixel locations in the original image, these are transformed into locations given by $\hat{\boldsymbol{x}}$,

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{s}(\boldsymbol{x}). \tag{2.31}$$

The greyscale value associated with each location is moved accordingly, and so the image is transformed.

The pre-processing is now defined in terms of a gradient dynamics which adjusts $s(x)$, so as to minimise the distance between the transformed image and the prototypes. The pre-processing dynamics are derived from a potential function, which is based on the Euclidean distance between two images, such that the potential is zero when the test and prototype images are identical. In the case of small, local deformations [40, 16, 17], a cost term is added to penalise the system for large deformations.

Using a cost function based on the values of the dilation and shear forces required to produce the deformation on an isotropic elastic sheet, this system has been used successfully to recognise hand-written characters which were incorrectly classified using the standard approach.

2.4 SCAP as an Orthogonal Projection Method

The derivation of SCAP started with Haken's studies of synergetic behaviour in lasers, was developed into a form of pattern recognition using a synergetic dynamics and ended with the understanding that the dynamics were unnecessary for classification because the final state of the system could be predicted from the initial conditions.

SCAP can also be derived more directly from consideration of orthogonal, symmetric projection operators. This earlier, parallel derivation is the work of Noguchi [79], which has not previously been cited within the literature on synergetics.

Among other results, Noguchi proves that the order parameters defined by Equation (2.21) maximally reconstruct the test vector q over the subspace spanned by the prototypes. This can be expressed mathematically as,

$$||V\xi - q||^2 \leq ||V\beta - q||^2. \tag{2.32}$$

To prove this hypothesis, we look for the β which minimises the functional form,

$$\begin{aligned} \phi(\beta) &= ||V\beta - q||^2 \\ &= \beta^T V^T V\beta - 2\beta^T V^T q + q^T q \\ &\geq 0. \end{aligned} \tag{2.33}$$

Differentiating $\phi(\beta)$ with respect to β and setting all of the elements to zero yields,

$$V^T V\beta - V^T q = \mathbf{0}. \tag{2.34}$$

Re-arranging to solve for β, we find that,

$$\begin{aligned} \beta &= (V^T V)^{-1} V^T q \\ &= V^+ q \quad \text{(Equation (2.18))} \\ &= \xi. \end{aligned} \tag{2.35}$$

Therefore, the order parameters provide the optimal reconstruction of a test image over the space spanned by the prototypes.

By adding the restriction that each class must have the same number of training images, Noguchi also derived an alternative to MELT using the theory of orthogonal projection operators.

2.5 Conclusions

This chapter traces Haken's derivation of a synergetic pattern recognition scheme from an understanding of the physical processes that occur in synergetic systems. For a solid understanding of future chapters, there are two key points to be gained from this derivation. First is the concept that pattern recognition can be thought of as a form of pattern formation, where a given pattern is changed such that it forms a pattern previously memorised by the system. This is the key concept behind synergetic pattern recognition. Second is the idea that while the dynamics of such a pattern formation involves a huge number of dimensions, the evolution is in fact, completely controlled by the order parameters. This concept is important as it allows us to derive a low-dimensional system which is completely equivalent to the high-dimensional system, yet makes synergetic pattern recognition computationally inexpensive.

We have also reviewed the various synergetic pattern recognition algorithms which have been proposed in the literature. From this review it is clear that synergetic pattern recognition is a new field, with many unexplored options. In terms of developing practical pattern recognition systems, perhaps the most exciting element of synergetic pattern recognition is the fact that the potential surface, in contrast to many neural networks, has no spurious memories. It is also easy to use, in that one does not need to define a network structure and the learning algorithm is deterministic, again in contrast to many neural networks.

There are, however, a number of facets in which synergetic pattern recognition needs to be improved. Foremost among these is the classification power of the system, which is the subject of the next chapter.

Chapter 3

Enhanced Synergetic Pattern Recognition

3.1 Motivation

In Chapter 2, we introduced the original synergetic pattern recognition scheme along with several extensions proposed by various authors to cope with weaknesses in the original formulation. The sum of these algorithms constitutes the state of the art in synergetic pattern recognition.

Wang et al [103] recognised that probably the most limiting problem faced by someone attempting to use standard synergetic pattern recognition to solve a practical problem is the *inflexibility* of the decision boundaries. Having selected prototypes, the user has no further control of the system. Assuming that a test set reveals unacceptable error rates, the user has no option but to select a new set of prototypes and try again, because the decision boundaries are fixed.

In comparison, the user of a standard back-propagation type neural network can train the network with emphasis on the test data which is near the decision boundary, and this non-uniform training will change the position of the boundary. In a similar vein, Wang and his co-workers made the attention parameters variable, and introduced a training scheme to find optimal values for the attention parameters. Unfortunately, their training scheme is slow and cannot be guaranteed to take advantage of the extra flexibility found in the new system. Also, the pattern recognition times are much slower than those found using SCAP, because the system requires the evolution of a dynamic system.

In this chapter, we describe two new synergetic algorithms which allow the user to train the system to find optimal decision boundaries, and yet maintain the high classification speed of SCAP.

3.2 SCAPAP

The SCAP formulation of synergetic pattern recognition places much weight on the speed of the classification at the expense of the flexibility of the classification boundaries, and hence possibly on the accuracy of the classification. Wang et al [103] pro-

posed a closely related synergetic classifier with more flexible boundaries and showed that their system can indeed lead to increased classification accuracy. Unfortunately, their system suffers from much longer classification times and has an unsatisfactory training regime.

In this section we introduce a new algorithm which matches the discrimination capabilities of Wang's system, yet has the same time requirements as SCAP, and can be trained using linear programming.

We demonstrate our new algorithm on its ability to recognise pose, or the angle at which an object is presented to the camera. Irrespective of their mathematical implementation, pose recognition algorithms can be split into two streams, depending on the form of output produced. The first stream attempts to find a vector, $\boldsymbol{\theta}$, which captures the angles through which a given object has been rotated [19, 23, 14, 43]. The second classifies an object into *aspects* [35, 62, 78, 81, 95]. An aspect is a contiguous set of $\boldsymbol{\theta}$ values which is determined uniquely by the shape of the object.

Our approach allows us to recognise that a compromise between these two streams has applications in robotic manipulation. Here we classify objects into aspects that can be defined by the user. If a robotic arm has several different grasps, depending on the pose of an object, the user-defined aspects are an effective way for the robot to decide which grasp is required [30].

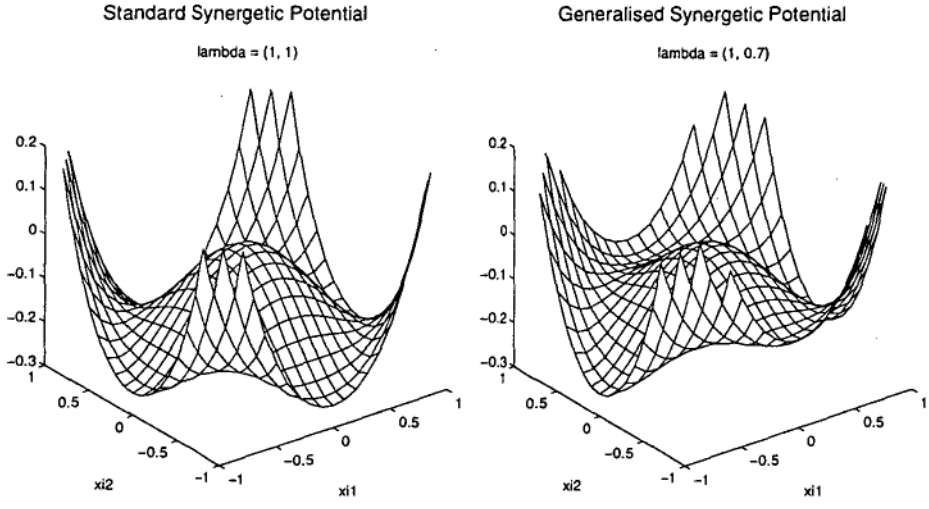
The characteristics of our technique are well suited to the nature of the manufacturing environment. First, our algorithm classifies quickly and requires only the use of a standard video signal. Thus it is completely *passive* and need not interfere with, or slow down, the manufacturing process. Second, since the external variables can be highly controlled, we can assume that translation, scale, lighting conditions and camera parameters are all fixed. The appearance of the object will therefore be dependent purely on the shape and pose of the object.

Motivated by the need for a more accurate and flexible approach, we discuss the need for generalisation of the classification model in Section 3.3, and introduce an extension based on freeing the so-called attention parameters [103]. Analysis of the extended model leads to a new, more powerful algorithm which we label SCAPAP, standing for SCAP with attention parameters. Section 3.4 describes a deterministic training procedure for the new classification scheme and presents some examples of pose classification which demonstrate the increased power of the new technique.

3.3 Generalisation

The decision boundary used to separate classes k and l in the SCAP algorithm is defined by the surface, $\xi_k = \xi_l$. The boundary is therefore both linear and fixed. The success of SCAP is reliant on the test images being projected correctly into order-parameter space, and the only course of action available if this does not succeed is to select new prototypes and start again.

By loosening some of the restrictions given by Equation (2.24), we aim to make the decision boundaries more flexible. When testing shows an unsatisfactory error rate, the system can then be tuned using the free parameters, to minimise the error over a training set.



(a) The standard synergetic potential with $\lambda = \lambda = 1$.

(b) The generalised synergetic potential with $\lambda = (1, 0.7)$.

Figure 3.1: The standard synergetic potential is a special case of the generalised potential.

We choose to loosen the restriction on the attention parameters, such that,

$$\begin{aligned} B_{kl} &= c, \\ \lambda_k &> 0. \end{aligned} \tag{3.1}$$

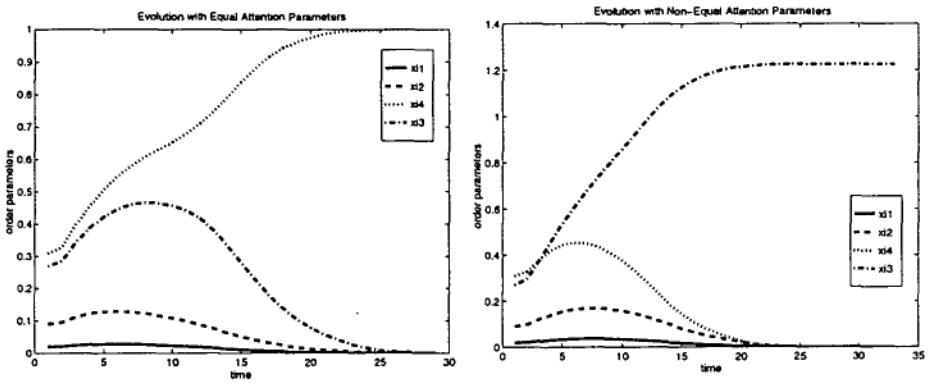
We can see the effect this has on the potential surface V_{AP}, where the subscript AP denotes 'attention parameters', in the two-dimensional case shown in Figure 3.1(b). In this instance, all four minima are accessible, but the two minima associated with class 1 have larger basins of attraction than those associated with class 2. This figure is in contrast to Figure 3.1(a) which shows the restricted potential surface which is the basis of SCAP.

The generalised pattern formation model PF_{AP} is given by,

$$\dot{\xi}_k = f_k(\boldsymbol{\xi}, \boldsymbol{\lambda}) = \xi_k(\lambda_k + c\xi_k^2 - 2c\sum_{l=1}^{n}\xi_l^2). \tag{3.2}$$

We already have an intuitive feel for the effect of changing the attention parameters as they control the depth of the minima on the potential surface. Also, choosing to free the attention parameters has a number of advantages over freeing the competition parameters, B_{kl}. First, the analysis is made easier by removing the summation in the second term of Equation (2.23). Second, the number of free parameters grows linearly, not quadratically, with the number of classes.

The new system, PF_{AP}, is capable of behaviours not seen in PF_R. An example of this can be seen by contrasting the evolution of PF_{AP} in Figure 3.2(b) with the evolution of PF_R in Figure 3.2(a), using the same initial conditions.



(a) PF_R (b) PF_{AP}

Figure 3.2: The final state of the order parameters determines the classification.

Note that for this particular set of attention parameters, the PF_{AP} model classified the pattern as belonging to class 3, where the PF_R model must select class 4. Control of the attention parameters therefore allows the system designer to control the location of the class boundaries.

3.3.1 Analysis

The classification boundary of the PF_R model is simple and can therefore be reproduced exactly by the SCAP algorithm, but the model itself is restrictive in the classes that it can separate. In adopting the PF_{AP} model, our goal is to approximate the more complicated boundary so that we can use the extra descriptive power of the model, while maintaining the speed of the linear algorithm.

3.3.2 The Final States

Equating the right-hand side of Equation (3.2) to zero, it is clear that the PF_{AP} model has 2^n stationary points consisting of both zero and non-zero elements. A solution with $m \leq n$ non-zero elements is denoted as ${}^m\xi$ and given by,

$$ {}^m\xi = \begin{cases} \xi_k^2 = 2\sum_{j=1}^{n} \xi_j^2 - \frac{\lambda_k}{c}, & k = 1, \ldots, m \\ \xi_k = 0 & k = m+1, \ldots, n. \end{cases} \tag{3.3} $$

Wang et al. [103] have shown that the potential energy achieves its minimum value when $m = 1$. Thus the global minima correspond to the prototype images, which are projected onto,

$$ {}^1\xi = (0, \ldots, \sqrt{\frac{\lambda_k}{c}}, \ldots, 0), \tag{3.4} $$

in order-parameter space. We can conclude, therefore, that all prototypes are stable final states.

As a classification model, it is important that all prototypes are possible final states of PF_{AP} simultaneously. The linear stability matrix, A, for Equation 3.2 is given by

$$A = \left(\frac{\partial f_i}{\partial \xi_j}\right), \tag{3.5}$$

where

$$\begin{array}{rcl} \frac{\partial f_k}{\partial \xi_k} & = & \lambda_k - 2c\sum_{i=1, i\neq k}^{n} \xi_i^2 - 3c\xi_k^2 \\ \frac{\partial f_k}{\partial \xi_l} & = & -4c\xi_k\xi_l \end{array}. \tag{3.6}$$

Substituting the kth prototype solution (Equation 3.4) into the stability matrix gives,

$$A = diag(\lambda_1 - 2\lambda_k, \ldots, -2\lambda_k, \ldots, \lambda_n - 2\lambda_k), \tag{3.7}$$

which is stable only if all of the diagonal terms are negative. Since we require stability for all prototypes simultaneously, we must choose the attention parameters to satisfy,

$$2\lambda_i > \lambda_k > \frac{1}{2}\lambda_i \qquad \forall i \neq k. \tag{3.8}$$

Stability analysis of the origin ($m = 0$) and local minima ($m > 1$) shows that these stationary points are all unstable [103]. We can summarise these results by stating that,

- all the prototypes are simultaneous possible final states given restrictions on the choice of the attention parameters (Equation 3.8); and
- there are no other possible final states.

The location of the stationary point ${}^n\boldsymbol{\xi}$ is fundamental to the behaviour of the system. We label it $\boldsymbol{\xi}^*$ and state that it is given by,

$$\xi_k^* = +\sqrt{\frac{1}{c}\left(-\lambda_k + \frac{2}{2n-1}\sum_{i=1}^{n}\lambda_i\right)}, \tag{3.9}$$

which can be verified by substitution into Equation (3.2). Now substituting Equation 3.8 into Equation 3.9 demonstrates that ξ_k^* is real and positive.

Rearranging this equation yields the relationship,

$$\sum_{i=1}^{n}\lambda_i = c(2n-1)\sum_{i=1}^{n}\xi_i^{*2}, \tag{3.10}$$

which can then be used to define λ_i in terms of the location of ξ_k^*,

$$\lambda_i = 2\sum_{i=1}^{n}\xi_i^{*2} - c\xi_i^{*2}. \tag{3.11}$$

3.3.3 Predicting the Final State

2-Class Classification

We now look at the case where there are two user-defined aspects, which is to be implemented as a 2-dimensional version of Equation (3.2).

Fortunately, this system is amenable to phase-plane analysis, which allows us to predict the long-term behaviour of the evolution, as we have shown previously [44]. The system has four stationary points: the origin, which is unstable, the two prototype solutions (Equation 3.4), which are stable if Equation 3.8 is satisfied, and ξ^*. Substituting $n = 2$ into Equation 3.9 gives,

$$\begin{aligned} \xi_1^* &= \sqrt{\tfrac{2\lambda_2 - \lambda_1}{3c}}, \\ \xi_2^* &= \sqrt{\tfrac{2\lambda_1 - \lambda_2}{3c}}, \end{aligned} \tag{3.12}$$

and the stability matrix (Equation (3.5)) yields two eigenvalues,

$$-\lambda_1 - \lambda_2 \pm \sqrt{-23\lambda_1^2 + 62\lambda_1\lambda_2 - 23\lambda_2^2}. \tag{3.13}$$

Because of the restrictions of Equation (3.8), the eigenvalues have opposite signs, and so ξ^* is a saddle point.

The nature of the stationary points allows us to state that there is a separatrix between the origin and (ξ_1^*, ξ_2^*) which defines the boundary between the two classes. SCAPAP approximates this boundary by the line segment between these points and classifies patterns projected below the line as Class 1, and those above as Class 2.

This analysis has been tested numerically for a number of choices of the attention parameters, as shown in Figure 3.3. Each graph shows the boundaries between classes for SCAPAP and the generalised pattern formation model PF_{AP} for a particular attention parameter set. The boundaries were found by classifying each point in order parameter space using the two different methods and then dividing order parameter space accordingly.

Figure 3.3 shows good agreement between the SCAP boundary and the separatrix predicted using phase-plane analysis. Beyond the saddle point, however, the analysis does not predict the shape of the class boundary and the straight-line approximation is poor except in the special case of equal attention parameters. In Section 3.3.4, we will show for the general n-dimensional case, that no images are projected into this region of space.

The SCAPAP Algorithm

We can scale order-parameter space so that the axes are now defined by

$$x_i = \frac{\xi_i}{\xi_i^*}. \tag{3.14}$$

The barrier between the two classes is now the line $x_1 = x_2$ and, in the same manner as SCAP, SCAPAP classifies by selecting the class with the largest initial value among the x_k's.

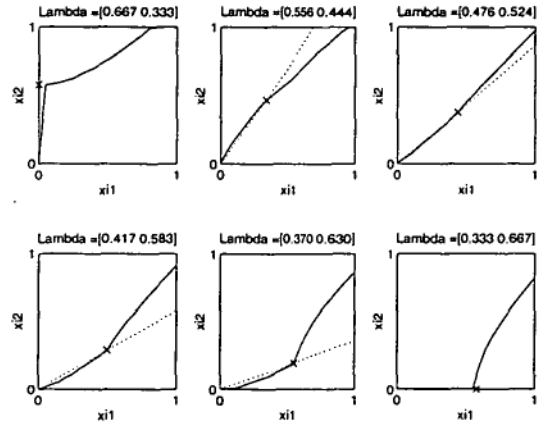


Figure 3.3: 2-dimensional SCAPAP boundaries. Graphs show the actual boundary (solid line), approximate boundary (dashed line) and saddle point (x).

n-Class Classification

We have shown above that the only stable solutions to the full n-dimensional PF_{AP} are the prototype patterns. Unfortunately, the question of predictability cannot be solved in higher dimensions using phase-plane analysis.

We therefore use a numerical approach to find the boundaries in higher dimensions. In a natural extension of n-dimensional SCAP and 2-dimensional SCAPAP, we conjecture that the border between classes i and j is defined by the surface,

$$x_i = x_j. \tag{3.15}$$

We could test this conjecture for a specific problem by acquiring a very large number of images and comparing the predicted result to the result of the full dynamic system. However, such a large image set would be very expensive to obtain, and the conclusions difficult to generalise.

Instead of following this approach, we looked at order-parameter data from a number of different problem sets, and modelled the distribution of the order-parameter sets as normal distributions. Using the resulting approximations for the mean, μ, and standard deviation, σ, it was possible to artificially create very large order-parameter data sets of 10,000 samples inexpensively. Table 3.1 shows the percentage of the 10,000 samples for which the SCAPAP classification agreed with that given by the pattern formation model, PF_{AP}. Each row represents a certain number of classes to be separated, ranging from 2 to 10. For each dimension, 10 systems with randomly selected attention-parameter sets were tested, and the columns of the table show the minimum, maximum and mean percentage agreement over the 10 systems.

We believe the tabulated values are pessimistic for two reasons. First, the percentage correctly classified by the conjectured border decreased with increasing standard deviation. We chose a mean of $\mu = .15$ and a standard deviation of $\sigma = 0.2$, which was the largest standard deviation from the problem sets. Second, the original data on which the normal distribution was modelled showed a better agreement than the

Dimension	min	max	mean
2	96.2	99.9	96.9
3	95.6	98.3	97.1
4	85.9	96.4	90.9
5	80.6	92.0	85.7
6	80.3	87.0	84.2
7	75.9	93.1	82.3
8	78.6	83.6	81.2
9	78.3	82.7	80.2
10	73.7	78.6	76.9

Table 3.1: Percentage of space correctly classified by SCAPAP.

large sample sets. This is probably associated with the dependence between the order parameters that was not captured in the univariate, normal-distribution model.

3.3.4 The Initial States

Figure 3.3 shows that there is a region beyond ξ^* in which the SCAPAP boundary is a poor approximation to the actual boundary. We wish to construct a system in which no image q is projected into this problematic region of order-parameter space.

Theorem 1 (Initial States Theorem.) When the attention parameter set is chosen such that,

$$\sum_{i=1}^{n} \xi_i^{*2} \geq 1 \tag{3.16}$$

no image is projected into the problematic region.

The Initial States Theorem is proved in Appendix A.

3.4 Training

The PF_{AP} model has $n+1$ free variables which can now be chosen to minimise the classification error over a training set. As the constant, c, scales all of the variables equally, we now set $c = 1$ without loss of generality.

3.4.1 Award-Penalty Learning

In contrast to many neural-network systems, the relationship between the free variables and the expected SCAPAP classification is understood. It is clear from Equation (3.2) that an increase in ξ_k will increase the likelihood that a test image is classified as belonging to class k. We can use this knowledge to adjust the free variables using an award-penalty learning mechanism. This training paradigm alters the parameters

whenever a test pattern is incorrectly classified. The attention parameter corresponding to the correct result is 'awarded' an increase by (a small, user-defined) δ. At the same time, the attention parameter for the incorrect result is 'penalised' by a decrease of δ. These two adjustments will increase the probability of the correct result's being found next time the image is classified. The parameters are left unchanged whenever an image is correctly classified.

One run through the training set is called a *training epoch.* The user sets an acceptable number of errors for a single epoch and the training continues until the number of classification errors is acceptable.

Wang et al. [103] successfully used this mechanism to train the attention parameters of Equation (3.2) for an optical character-recognition problem. However, this approach has a number of weaknesses.

First, the final trained system cannot be guaranteed to be an optimum solution on the training set. The method has a number of user-defined parameters, namely: the award/penalty value, δ; the maximum allowable errors; and the initial attention parameter set. All of these will affect the final parameter set.

Second, the training time is likely to be lengthy, because every training image must be classified once for every training epoch. Previously, this involved integrating Equation (3.2), but the results of Section 3.3.3 allow us to replace each of these integrations with the SCAPAP algorithm.

Even given this reduction, the time required to train the system is indeterminate. Indeed, the training may cycle indefinitely depending on the user-defined parameters.

3.4.2 Explicit Parameter Learning

Here we develop a training system for the PF_{AP} model which takes advantage of the classification criterion of SCAPAP to avoid the problems described above.

Each element of the training set defines $n-1$ inequality relationships between the attention parameters. Because we now understand the implications of adjusting any given attention parameter, we can move from the 'blind' training technique described above, to one that will always give an optimum result in a single training epoch.

For an n-dimensional SCAPAP system, we want to find $\lambda_{1\ldots n}$ such that the system classifies the training set with minimum error.

We can simplify the algebra of this problem by expressing it in terms of finding the optimum ξ^*. We then use Equation (3.11) to find the corresponding optimum attention-parameter set.

Consider a training set T of m images, q_i, and a correct classification for each image $\gamma_i \in \{1, \ldots, n\}$. Now for each (q_i, γ_i) pair, SCAPAP requires that,

$$(\gamma_i = k) \Rightarrow (x_k > x_l) \qquad \forall l \neq i \tag{3.17}$$

for the image to be correctly classified.

This gives $m(n-1)$ restrictions of the form,

$$\xi_k * < \frac{\xi_k}{\xi_l} \xi_l^*. \tag{3.18}$$

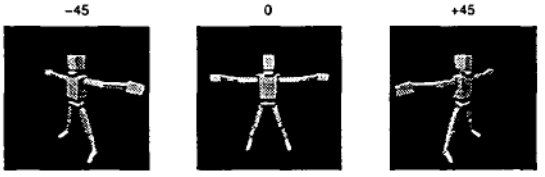


Figure 3.4: 'Man' rotated at $-45°$, $0°$ and $+45°$.

We can now find a solution to these restrictions simultaneously using an n-dimensional linear program. To complete the list of restrictions, we add n inequalities to ensure that the saddle point is real and positive,

$$\xi_k* > 0, \qquad k = 1, \ldots, n, \tag{3.19}$$

and the restriction required by the Initial Values Theorem,

$$\sum \xi_i^{*2} \geq 1. \tag{3.20}$$

This final restriction cannot be used directly because of its non-linearity. We can, however, use the linear restriction,

$$\sum \xi_i^* \geq 1, \tag{3.21}$$

which implies the wanted restriction.

Now any feasible solution for ξ^* will correctly classify the entire training set.

3.4.3 2-Class Training Example

A 2-class example is useful for visualising the training process. The goal of this example is to classify images of a single object into two user-defined aspects.

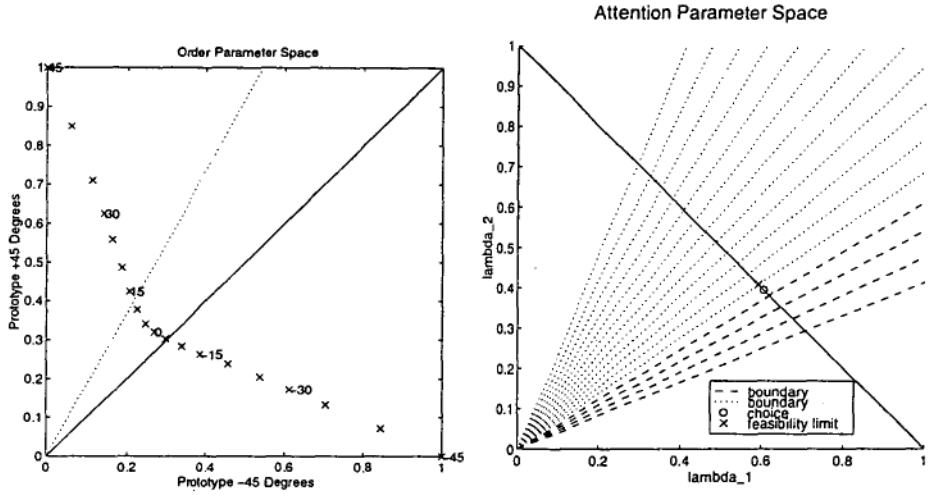
The data were a set of computer-generated images of a stick figure titled, 'man', rotated around the vertical axis. The rotation varied in 5°-steps from $-45°$ to $+45°$. The image at $0°$ showed 'man' looking straight at the viewer. Three of the images are shown in Figure 3.4.

The images at $\pm 45°$ were used as the prototypes for the SCAP algorithm. Figure 3.5(a) shows the smooth curve formed in order-parameter space when the complete set of images is projected into the space.

Using the SCAP algorithm, the Actual Decision Boundary has fallen naturally between $-5°$ and $-10°$. Using SCAPAP, users can decide where they want the decision boundary to lie. For this example, we want the boundary to lie between $+10°$ and $+15°$.

The training process is shown in Figure 3.5(b). We use all of the images as the training set although, in this case, we need only use the two nearest the boundary. Each of the images results in either a greater-than (dotted line) or less-than (dashed line) inequality relationship.

It can be seen from the Figure 3.5(b) that there is an infinite number of attention-parameter sets that will produce the required boundary. We choose to select one on



(a) Order-parameter space: Actual Decision Boundary (line) and Desired Decision Boundary (dotted line).

(b) Attention-parameter space.

Figure 3.5: 2-class training example.

the line $\lambda_1 + \lambda_2 = 1$. The upper and lower bounds of feasibility on this line are shown as crosses. We selected the point mid-way between these and marked it with a circle.

The selected attention-parameter set was then verified using the full dynamical system of Equation (3.2) and the 2-dimensional SCAPAP algorithm described in Section 3.3.3. In both cases the entire training set was classified correctly.

3.4.4 n-Class Training Example

As an example of training an n-class system, we present an extension of the 2-class system described above. Given an object which is rotated around one axis, we wish to divide the rotation domain into a set of n user-defined aspects.

In this example we use real images of a child's toy, rotated 360° around one axis in 5°-steps. We use four prototypes at 0°, 90°, 180° and 270°, which are shown in Figure 3.6. The images are part of the COIL database, which is described in detail by Nene et al. [77].

The desired classification is shown in Table 3.2,

We defined a training set T of images which consisted of the following images,

$$T = \{45, 50, 135, 140, 220, 230, 315, 320\}, \tag{3.22}$$

and trained the system using a standard linear programming code. The program converged to yield a value for ξ^* which satisfied all of the requirements described in Equation (3.18).

Having trained the system to find a value for ξ^*, we converted this into a set of attention parameters using Equation (3.11). We then used a test set which contained

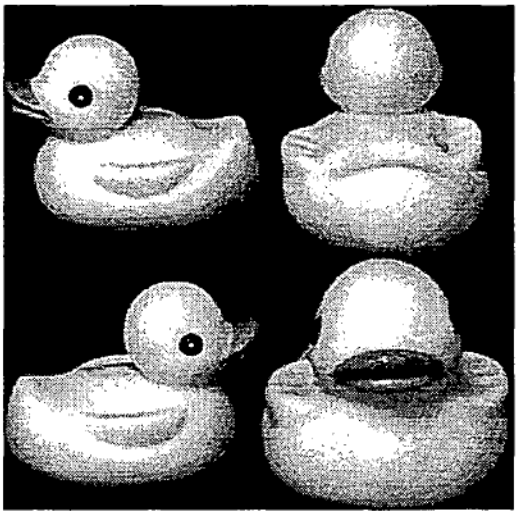


Figure 3.6: Prototype images of child's toy (from COIL database).

Rotation Range	Class
320 - 45	1
50 - 135	2
140 - 225	3
230 - 315	4

Table 3.2: Correct Classifications of Child's Toy.

all 72 images in the database as a basis of comparison between the pattern-formation models studied in this chapter. The results, which are a significant improvement over SCAP, can be seen in Table 3.3.

The fourth row is of particular interest. This is the error of the full non-linear system PF_{AP} when using the chosen attention-parameter set. In Table 3.1, the mean agreement between SCAPAP and PF_{AP} for a 4-class system was quoted as 90.9%. For this actual example, however, 70 of the 72 patterns were classified in the same way, which has 97% agreement. This is further evidence to suggest that the results of Table 3.1 using artificial order parameter data is pessimistic in comparison to real image data.

Note also the disappointing result in the second row of Table 3.3 for the award-penalty training scheme used by Wang et al. [103]. For this dataset, the initial value $\lambda = [1, 1, 1, 1]$ is a local minimum and so the training scheme is incapable of improving on the results returned by SCAP.

Algorithm	Train	Test
SCAP	-	11
Wang et al.	4	11
SCAPAP	0	1
PF_{AP}	-	2
SCAPAP-P	0	0

Table 3.3: Classification Error.

3.5 SCAPAP-P

As the size of the training set and the number of classes increases, the likelihood that a feasible set of SCAPAP attention parameters will exist decreases. This is because the single set of chosen attention parameters must satisfy all of the increasing number of restrictions simultaneously.

SCAPAP-P is a variation on SCAPAP, where the P stands for parallel, and denotes the parallel nature of the algorithm. SCAPAP-P uses the same generalised pattern formation model, PF_R, and uses the same linear approximation to the non-linear decision boundaries as SCAPAP. The distinction is that for an n-class scheme, SCAPAP-P consists of n 2-class SCAPAP systems in parallel. Each of these subsystems has been trained to decide if a test image belongs to a given class or not.

In this approach, the SCAPAP subsystems require a different training scheme which reflects their new, simpler role. For each subsystem we find a separate set of attention parameters, which we will distinguish with a superscript, $\boldsymbol{\lambda}^k$, which distinguish only between belonging to class k and not belonging to class k. This set is not intended to distinguish between any other classes.

It is simple to write down the restrictions required for this training. To train the kth set, $\boldsymbol{\lambda}^k$,

$$\begin{array}{ll} x_k > x_i & \gamma_i = k \\ x_k < max(x_i) & \gamma_i \neq k. \end{array} \tag{3.23}$$

The advantage of SCAPAP-P over SCAPAP is the extra flexibility gained from having multiple sets of attention parameters. By dividing the task of n-class classification into n 2-class classification problems we have increased the likelihood that the final classification will be successful. Indeed, in Table 3.3 it is shown that SCAPAP-P was capable of completely separating the four classes as required.

The drawback is that the restrictions of Equation (3.23) are non-linear, so such a system cannot be solved using linear programming techniques. However, the relative simplicity of the requirements allowed us to use a simple Monte Carlo search to find appropriate attention parameter sets for this example.

3.6 Conclusions

SCAPAP represents a significant improvement over the current generation of synergetic pattern recognition algorithms. The generalised pattern formation process at the heart of SCAPAP yields a more flexible boundary than that provided by SCAP, and yet the classification times for the two algorithms are equal.

The classifications made by SCAPAP approximate those made by the algorithm of Wang et al [103], yet in comparison the classification times required by SCAPAP are a major reduction, as we do not need to allow a system of differential equations to evolve.

Furthermore, we have shown that the attention parameters used by both SCAPAP and Wang et al, can be trained using linear programming. This provides a deterministic, non-iterative scheme which does not rely on any arbitrarily defined user parameters and is guaranteed to return a set of attention parameters with zero error over the training set, if such a set exists.

In the case that no feasible set of attention parameters exist, we have the option to use SCAPAP-P. This algorithm increases the likelihood of success by creating a number of SCAPAP systems in parallel, each of which has simpler requirements than the original SCAPAP system.

Chapter 4

Synergetic Pattern Rejection

4.1 Motivation

The standard synergetic recognition model [40] is incapable of concluding that a novel image does not belong to any of the learned classes. This inability to *reject* an image is a serious weakness of the approach as even white noise, or a novel image which bears no resemblance to any of the prototypes, will be classified as belonging to the same class as one of them. In this chapter we start by introducing the concept of a rejection threshold. It is then a simple matter to add a rejection rule to SCAP, stating that any order parameters falling below their respective rejection threshold will be set to zero. While this method results in a practical linear classification scheme capable of rejection, the scheme is no longer derived from the synergetic potential. This is aesthetically unpleasing because the link to pattern formation has been broken, and it has real implications in the attempts to produce physical computing devices capable of synergetic pattern recognition [37]. We therefore introduce instead, an extension to the standard synergetic potential which leads to a dynamics in which an image can be rejected. The resulting classification system can be tuned by the user to allocate desired rejection threshold values for each class.

4.2 Rejection Threshold

Consider a classification scheme with a single prototype representing each class. Given a test image, we can calculate a measure of its similarity to each of the prototypes and decide to which class the test image belongs. If we want to be able to reject an image as belonging to none of the classes, the obvious approach is to set *rejection thresholds* for each class. A similarity measure below the rejection threshold signifies that the image cannot be classified to that class. In the case that each of the similarity measures fall below their respective rejection thresholds, the image is rejected.

In general, rejection thresholds are problem specific and can only be set after an extensive trial and error procedure. This is due to the fact that the rejection thresholds must vary with the similarity between the prototypes. As discussed by Böbel et al. [12], we can find analytic upper bounds on rejection thresholds for SCAP in the two class case.

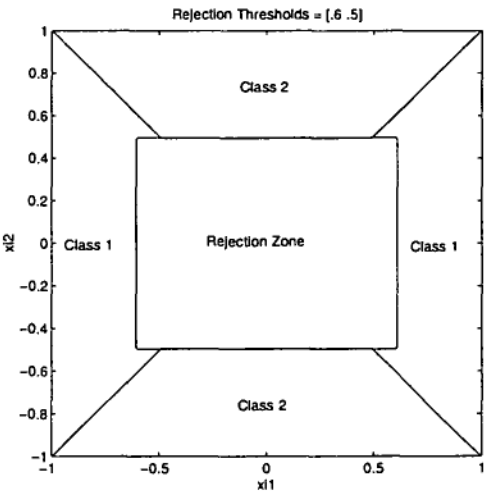


Figure 4.1: The classification and rejection boundaries using simple rejection thresholds.

To see how this is done, we proceed by finding the inter-class boundary for SCAP, given by

$$\xi_1 - \xi_2 = (\boldsymbol{v}_1^+ - \boldsymbol{v}_2^+)\boldsymbol{q} = 0. \tag{4.1}$$

An upper bound on the rejection threshold for class 1, $\hat{\xi}_1$, is given by the value of ξ_1 for which membership of class 1 is assured. This can be expressed mathematically as,

$$\hat{\xi}_1 - \boldsymbol{v}_2^+\boldsymbol{q} > 0 \qquad \forall \boldsymbol{q}. \tag{4.2}$$

Now as shown in Equation (2.17), all test images, $\boldsymbol{q}$ can be expressed as a linear superposition of the prototypes plus a small term orthogonal to the prototypes which plays no part in the classification process.

Substituting this expression for $\boldsymbol{q}$ into Equation (4.2) and enforcing the orthonormality between the prototypes and adjoint prototypes leads to,

$$\hat{\xi}_{1,2} \leq \frac{1}{\sqrt{2(1+\boldsymbol{v}_1\boldsymbol{v}_2)}}. \tag{4.3}$$

We can find a problem-independent upper bound by assuming the smallest possible value of $\boldsymbol{v}_1\boldsymbol{v}_2$ as equalling zero. This reveals that for all possible images the rejection threshold should be less than $1/\sqrt{2}$.

Figure 4.1 shows how a simple rejection thresholding can be used to introduce a subset of order parameter space in which images are rejected. This two-dimensional case can be simply extended to the general n-dimensional case by a hypercube centred on the origin with arbitrary lengths on each axis.

The approach of using simple rejection thresholds therefore offers a practical, linear approach to synergetic image recognition.

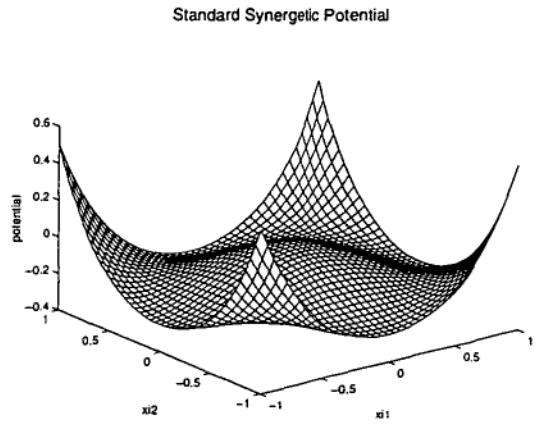


Figure 4.2: Standard synergetic potential.

4.3 A Synergetic Rejection Potential

In Figure 4.2 we see the standard potential for a 2-class problem, where the origin represents a rejection of all possible classes. While the origin is a stationary point, it is clearly unstable and will never actually be reached. This surface is defined by the standard restricted synergetic potential used as the basis of SCAP and given by Equation (2.24) as,

$$p_R = -\frac{1}{2}\lambda\sum_{k=1}^{n}\xi_k^2 + \frac{1}{4}(b+c)\left(\sum_{k=1}^{n}\xi_k^2\right)^2 - \frac{1}{4}b\sum_{k=1}^{n}\xi_k^4. \tag{4.4}$$

We now add an extra term to the potential which will place a minimum at the origin and leave the remaining section of the potential as unchanged as possible. We model the potential well as,

$$p_{well} = \left[\sum_{k=1}^{n}\left(\frac{\xi_k}{\sigma_k}\right)^2 - 1\right] e^{-\frac{1}{2}\sum_{k=1}^{n}\left(\frac{\xi_k}{\sigma_k}\right)^2}, \tag{4.5}$$

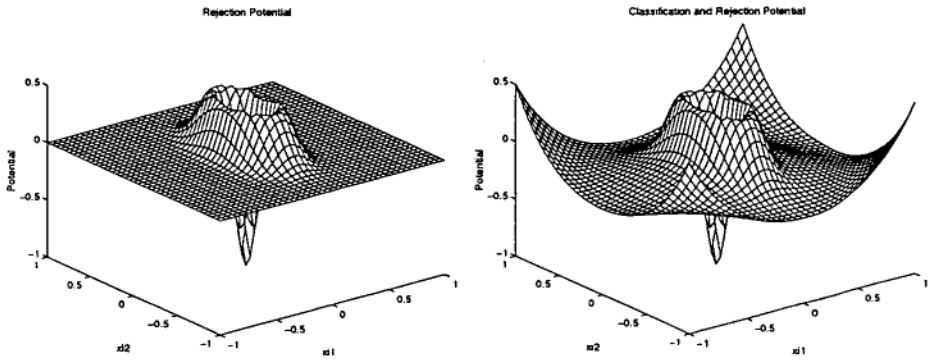
where σ_i is a variable that controls the length of the potential well along the i-th axis.

We chose this function for a number of reasons. First, it has a minimum at zero and approaches zero from the positive side, as $|\xi|$ becomes large. Second, it is differentiable. Third, it can be easily extended to an arbitrary number of dimensions.

A two-dimensional example of this potential well can be seen in Figure 4.3(a). In this instance, the well was created with $\sigma_1 \neq \sigma_2$.

The new synergetic potential with rejection well, formulated by $p = p_R + p_{well}$, can be seen in Figure 4.3(b).

Now that we have a formulation for the new potential, we need to choose σ_i such that the boundary between rejection and classification occurs at a point defined by the user. From Figure 4.3(b) it is clear that this boundary crosses each of the axes and is



(a) Rejection Well (b) Potential with Rejection Well

Figure 4.3: Construction of the new potential.

formed by a maximum in the potential surface. Any image projected to a point inside this boundary will be rejected. Any image projected to a point outside the boundary will be classified. Say that for each class we have determined a value, $\hat{\xi}_i$, at which this boundary should occur on the ith axis. We require that the potential have a maximum here, so we can express this as,

$$\frac{\partial p}{\partial \xi_i}(0, \ldots, \hat{\xi}_i, \ldots, 0) = 0. \tag{4.6}$$

Differentiating the new potential,

$$\frac{\partial p}{\partial \xi_i} = \xi_i \left[-\lambda + (b+c) \sum_{k=1}^{n} \xi_k^2 - b\xi_i^2 + \frac{1}{\sigma_i^2} \left(-\sum_{k=1}^{n} \left(\frac{\xi_k}{\sigma_k} \right)^2 + 3 \right) e^{-\frac{1}{2}\sum_{k=1}^{n} \left(\frac{\xi_k}{\sigma_k} \right)^2} \right] = 0, \tag{4.7}$$

evaluating at the appropriate value of the order parameter vector and re-arranging leads to the following expression for σ_i in terms of known values,

$$\frac{1}{\sigma_i^2} \left[-\frac{\hat{\xi}_i}{\sigma_i} + 3 \right] e^{-\frac{1}{2}\left(\frac{\hat{\xi}_i}{\sigma_i} \right)^2} = \lambda - c\hat{\xi}_i^2. \tag{4.8}$$

We can now solve this numerically for σ_i to complete our definition of the synergetic rejection potential.

4.4 Understanding the Evolution

We have now defined a potential with a rejection border at certain known points along the axes of the system. Before using our extended potential function for image classification and rejection, we wish to confirm that the evolution on the potential surface acts as required. We proceed by breaking the problem into two sub-cases.

4.4.1 Equal Rejection Boundaries

Here we look at the special case when all classes are given the same rejection boundaries. In this case $\hat{\xi}_i = \hat{\xi}$ and therefore, by virtue of Equation (4.8), $\sigma_i = \sigma$.

Given these equalities, the surface has a high degree of symmetry which can be used to predict the results of the evolution.

Class-Rejection Boundary

First, we want to find exactly where the boundary of the rejection well lies. This occurs when Equation (4.7) is equal to zero for all i. Excluding the origin, the boundary $\bar{\xi}_i$ is given by,

$$-\lambda + (b+c)\sum_{k=1}^{n}\bar{\xi}_k^2 - b\bar{\xi}_i^2 + \frac{1}{\sigma^2}\left[-\frac{1}{\sigma^2}\sum_{k=1}^{n}\bar{\xi}_k^2 + 3\right] e^{-\frac{1}{2\sigma^2}\sum_{k=1}^{n}\bar{\xi}_k^2} = 0 \qquad \forall i. \quad (4.9)$$

Now given the symmetry of the system, and the fact that we have designed the boundary to pass through $(0,\ldots,\hat{\xi},\ldots,0)$, we now test the supposition that the boundary is the n-dimensional hypersphere of radius $\hat{\xi}$. We do this by substituting the equation of the hypersphere,

$$\sum_{i=1}^{n}\bar{\xi}_i^2 = \hat{\xi}^2, \quad (4.10)$$

into Equation (4.9). The resulting equation is the same as Equation (4.8), which we have made to be true by our choice of σ. We can therefore conclude that the class/rejection boundary for this special case is a hypersphere centred at the origin, with radius $\hat{\xi}$.

Inter-Class Boundary

When outside the rejection zone, the inter-class boundaries are the same as those found for SCAP. The proof of this follows using the same argument used by Haken [40] in finding the inter-class boundaries for SCAP. Using a gradient-descent based integration,

$$\dot{\xi}_i = \xi_i(a + b\xi_i^2) \qquad \forall i, \quad (4.11)$$

where

$$a = \lambda - (b+c)\sum_{k=1}^{n}\bar{\xi}_k^2 - \frac{1}{\sigma^2}\left[-\frac{1}{\sigma^2}\sum_{k=1}^{n}\bar{\xi}_k^2 + 3\right] e^{-\frac{1}{2\sigma^2}\sum_{k=1}^{n}\bar{\xi}_k^2}, \quad (4.12)$$

and is therefore consistent over all of the equations.

Now since Equation (4.11) is invariant to replacing ξ_i with $-\xi_i$, we can proceed assuming that $\xi_i \geq 0$ without loss of generality.

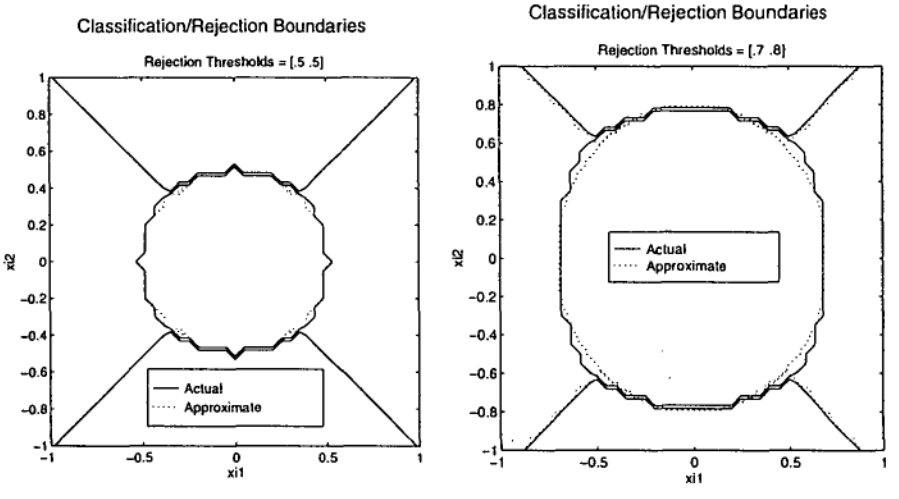
Assume without loss of generality that at time t, $\xi_i > \xi_j \ \forall i \neq j$, then

$$(a + b\xi_i^2) > (a + b\xi_j^2) \quad \forall i \neq j, \tag{4.13}$$

and, using Equation (4.11), we can state that $\dot{\xi}_i > \dot{\xi}_j$. Thus the largest initial order parameter will grow faster than any other order parameter and, due to the winner-takes-all nature of the system in the non-rejection area of parameter space, the final winner can be predicted by the order parameter with the largest initial absolute value,

$$\text{class} = \max_i |\xi_i(0)|. \tag{4.14}$$

We have run a number of numerical simulations to confirm our understanding of the system's evolution. To create Figure 4.4(a), which is a 2-dimensional example when $\hat{\xi} = [.5\ .5]$, we classified in two ways. First we integrated the full system of evolution equations until a minimum potential was reached. Second, we used our approximate classification scheme. In order to find the boundaries, we did this over the whole order-parameter space, and then found the location of the boundaries. The roughness of the circular rejection well seen in this image is due to the sampling of order parameter space. In fact, the boundary is completely smooth, and the approximate solution is very accurate. Note that this Figure also confirms that edges of the rejection well do occur at the rejection thresholds nominated by the user.



(a) Equal Rejection Thresholds: Class Boundaries

(b) Distinct Rejection Thresholds: Class Boundaries

Figure 4.4: Comparisons between Actual and Approximate Boundaries.

4.4.2 Distinct Rejection Boundaries

By allowing distinct rejection thresholds for each class, we break the symmetry of the potential surface which allows us to predict the results of the evolution precisely.

However, we know that the equal rejection threshold case is simply a special case of the general distinct threshold case, so the forms we use to approximate boundaries in the distinct rejection case should collapse to the boundaries found in the equal rejection case.

For the boundary between classification and rejection, we know that the boundary is symmetric around each axis, that it crosses each axis at the values of $\hat{\boldsymbol{\xi}}$, and that the completely symmetric case yields a hypersphere. Accordingly, our most reasonable approximation to this boundary is the hyper-ellipsoid,

$$\sum_{i=1}^{n} \left(\frac{\xi_i}{\hat{\xi}_i} \right)^2 = 1. \tag{4.15}$$

The inter-class boundary, which is defined by $\xi_2 = \xi_1$ in the equal rejection threshold case, is also affected by distinct rejection thresholds. We look for a boundary of the form, $\xi_2 = m\xi_1 + d$, which will collapse into the original boundary for the equal threshold case.

Now numerical simulation of the boundary suggests that for rejection threshold values required to separate realistic datasets such as those in the example below, the slope of the boundary, m is fixed at the value 1 while d varies such that the boundary approximately passes through $\hat{\boldsymbol{\xi}}$. The classification given by this approximate boundary is,

$$\text{class} = \max_i (|\xi_i(0)| - \hat{\xi}_i) \tag{4.16}$$

which clearly collapses to the equal threshold result when each element of $\hat{\boldsymbol{\xi}}$ is equal.

An example of this can be seen in Figure 4.4(b), which shows both the actual and approximate boundaries for the case $\hat{\boldsymbol{\xi}} = [0.8\ 0.7]$. The boundaries in this diagram were found in the same way as described above for Figure 4.4(a),

4.4.3 Example

To demonstrate the use of our synergetic classification/rejection potential, we now solve a two-class image recognition problem where some of the test data belongs to neither class. More specifically, our task is to classify or reject the images correctly, irrespective of the rotation angle at which they are presented to the camera.

In this experiment we have used three objects from the COIL database of images [77]. There are 72 images of each object, rotated around a natural axis of rotation in 5° steps. Figure 4.5 shows an example image of each of the objects.

We chose the duck and the wooden block to be the two classes and calculated a prototype for each class by combining four images of each object using the MELT algorithm. When all 216 images were projected onto the resulting order parameter space they were clustered, as can be seen in Figure 4.6. Superimposed on top of this projection are the boundaries found by our synergetic classification/rejection potential. The solid lines represent the boundaries found by the non-linear evolution of Equation (4.7) and the dotted lines represent the boundaries used by RSCAP, the

Figure 4.5: Three toys. The first two are examples of the two classes in the problem. The third is an example of an image that belongs to neither class.

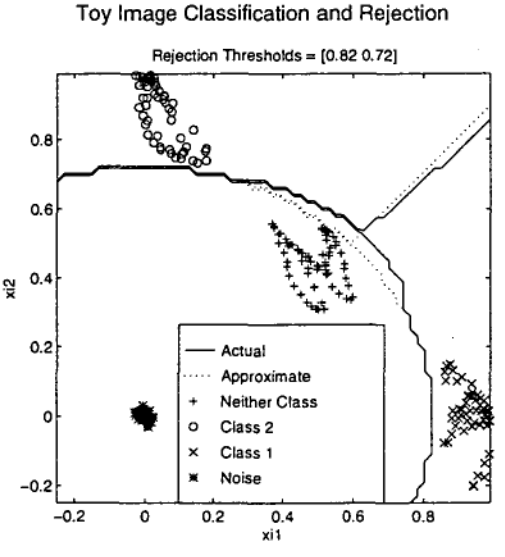


Figure 4.6: Order parameter space for the example problem.

linear approximation to those boundaries. It is clear that both sets of boundaries have been able to successfully reject the images that belonged to neither of the required classes. The cluster of points around the origin are the projections of images consisting entirely of noise. It is evident that very low rejection thresholds could be used to reject these inputs and hence ensure that non-significant input fluctuations remained unclassified.

4.5 Conclusions

The standard synergetic recognition potential is incapable of rejecting an image because the origin is an unstable solution. Any excitation whatever, no matter how small, will therefore cause such a system to evolve to its final state. Clearly this behaviour would be problematic in an attempt to create synergetic hardware.

In this chapter we have introduced an extension to the standard synergetic potential which creates a minimum at the origin. The resulting potential well, centred at the origin, is parameterised in an intuitive way, thus allowing the user to tune the system to reject white noise, or to reject real images which are projected below certain threshold values into order parameter space. Furthermore, analysis of the extended system shows that it behaves intuitively in terms of the location of boundaries.

For a practical, linear system capable of image rejection, the simple use of rejection thresholds is sufficient.

In terms of pattern recognition, the value of pattern rejection is clear. In the next chapter, we introduce the concept of synergetic learning. The role that rejection might play in this process is less clear, but it seems reasonable to assume that the human mind rejects some information that it cannot classify during learning. The application of this rejection potential to the task of synergetic learning is an interesting task for the future.

Chapter 5

Synergetic Learning

5.1 Motivation

Machines which make a decision based on a training set of examples must *learn* how to make that decision. Once the system has been *trained* on the training set, the machine is judged on its ability to *generalise* that knowledge successfully to novel inputs. Undoubtedly the most fundamental division of machine learning is into *supervised* and *unsupervised* learning.

In supervised learning, the training set of inputs has been labelled with the correct outputs. In this way the system knows when it has incorrectly categorised an input, and can attempt to modify its learning such that the entire training set is learnt correctly. The much quoted analogy is a classroom situation, in which a teacher provides the child with the correct answer.

To continue the analogy, unsupervised learning is learning without a teacher. The machine is simply given a number of inputs and is asked to group them into clusters of similar inputs. The user may give the machine a number specifying how many classes are present in the data, but no information as to which input belongs to which class. The machine must develop its own concept of similarity such that it can establish class boundaries and generalise those classifications to encompass novel inputs.

As an example-based classifier, synergetic pattern learning is a necessary precursor to synergetic pattern recognition, yet the learning process used by the synergetic classifiers described in Chapter 2 is implicit, rather than explicit. In this chapter we make it explicit by re-casting supervised synergetic learning in the familiar form of neural network learning so as to be able to draw parallels and highlight the distinctions between them.

We also review the current approach to unsupervised synergetic learning. The unsupervised learning procedure requires two dimensions for each pixel in the image, leading to an unwieldy and slow process. This massive dimensionality also results in the fact that the system cannot be guaranteed to reproduce the results of supervised learning without the introduction of two additional mathematical elements.

In contrast to this, we introduce our new form of unsupervised synergetic learning which is of a much lower dimension, is more robust and naturally reproduces the results of supervised learning.

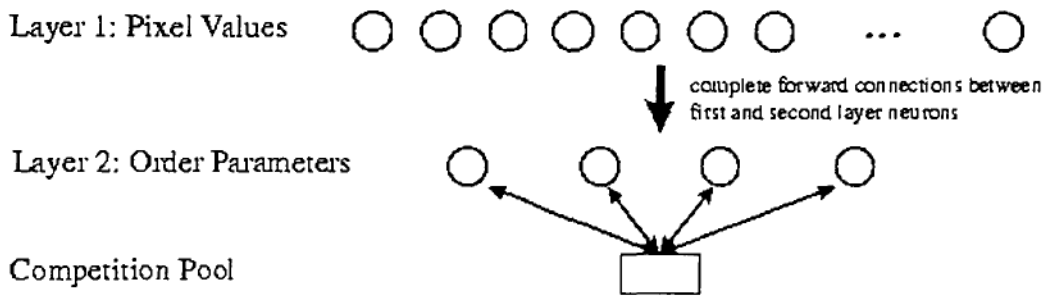


Figure 5.1: A neural network architecture that implements SCAP. Based on figure in [40].

5.2 Supervised Learning

All of the synergetic systems described so far, along with the majority of neural networks, use supervised learning. When discussing learning, it is illustrative to compare synergetic learning with the familiar neural network learning, so we follow Haken [40] and briefly re-formulate SCAP in a neural network architecture, as seen in Figure 5.1.

The design of a neural network requires decisions concerning the:

- structure and connectivity of the network;
- transfer function for each neuron; and the
- algorithm used to learn the synaptic weights.

We construct a two layer network in which the input layer has one neuron for each pixel and receives the greyscale values of the novel image, $\boldsymbol{q}(0)$ as input. Each neuron in the second layer is completely connected to the input layer. The activation of each second layer neuron is equal to the associated order parameter value, and is calculated by,

$$\xi_k = \sum_{i=1}^{\text{pixels}} V_{ki}^{+} \boldsymbol{q}_i(0). \tag{5.1}$$

So the second layer neurons sum the initial input as weighted by the connectivity matrix V^{+}. Now the role of the learning algorithm is to learn a connectivity matrix that minimises the error. As is suggested by our use of notation, we simply need to calculate the adjoint prototype matrix, as given by Equation (2.18).

So we have a deterministic procedure that directly calculates the connectivities. This is in contrast to the standard learning algorithms for neural networks, such as back-propagation, in which the parallel with the student/teacher analogy is much more clear. In general a neural network is started with a randomly chosen connectivity matrix. It is then presented with a series of training images and uses a learning

algorithm to fine tune the connectivity matrix and decrease the classification error. Training is stopped when the number of errors is acceptably low.

As an aside, the dynamical system can be implemented as a neural network by adding the 'competition pool', seen in Figure 5.1. This extra neuron allows the order parameters to compete for dominance in such a way that there is only one winner, with all other order parameters being suppressed.

5.3 Unsupervised Learning

For unsupervised learning, our task of finding the required connectivity matrix is unchanged. However, the goal of minimising the classification error is no longer valid, as we do not know the correct training image classifications. Instead we must introduce a potential function which is minimised when similar input patterns are classified together. By minimising this function over the training set, the system will develop its own categories, which can then be generalised in exactly the same manner as for a connectivity matrix constructed using supervised learning.

Fortunately in our case, we have access to just such a potential function already. In the previous chapters we have emphasised the fact that synergetic pattern formation is the basis of synergetic pattern recognition. It is also true that we can apply pattern formation principles to the *training* of synergetic systems [3, 40, 102, 36]. We can therefore construct synergetic systems which learn in an unsupervised environment using the same potential function, which we repeat here for the sake of convenience,

$$p = -\frac{1}{2}\sum_{k=1}^{n}\lambda_k(\boldsymbol{v}_k^+\boldsymbol{q})^2 + \frac{b}{4}\sum_{k\neq l}^{n}\sum_{l\neq k}^{n}(\boldsymbol{v}_k^+\boldsymbol{q})^2(\boldsymbol{v}_l^+\boldsymbol{q})^2 + \frac{c}{4}\sum_{k=1}^{n}\sum_{l=1}^{n}(\boldsymbol{v}_k^+\boldsymbol{q})^2(\boldsymbol{v}_l^+\boldsymbol{q})^2. \qquad (5.2)$$

The derivation of unsupervised learning described here is due to Haken [40]. The concept has been implemented in a neural network by Banzhaf and Haken [3] who successfully learned noisy & occluded images as well as demonstrating its relationship to Kohonen's self-organising maps [61]. It has also been implemented as a system of differential equations by Wagner et al. [102], who used their system to learn defects in woven materials.

For pattern recognition the adjoint prototypes $\boldsymbol{v}_k^+$ are fixed so the potential is a function of $\boldsymbol{q}$. In contrast, for unsupervised learning we hold the image $\boldsymbol{q}$ fixed and vary the adjoint prototypes $\boldsymbol{v}_k^+$ to find a minimum on the potential surface. Since we will in general have more than one image $\boldsymbol{q}$ to learn, we in fact minimise a sum of potentials, one for each of the m training images.

$$\hat{p} = \sum_{i=1}^{m} p(V^+, \boldsymbol{q}_i). \qquad (5.3)$$

Following the same approach as for pattern recognition, we can apply gradient dynamics to Equation (5.3), derive a system of differential equations and integrate the system until it converges on a connectivity matrix. Clearly, we would like the adjoint prototypes calculated in this way to stay in the space spanned by the training images during the integration, so they resemble the training images. Unfortunately, the

potential of Equation (5.2) does not enforce this. Rather than enforce this requirement at each time step, Haken [40] circumvented this problem by adding a new potential term which is minimal when the v_k^+ are in this subspace,

$$p_2 = \gamma|(1-f)q|^2 \tag{5.4}$$

Here γ is a decay constant and f is a projection operator defined by,

$$f = \sum_k v_k^+(t)v_k(t). \tag{5.5}$$

Now the final form of our potential is $\hat{p} + p_2$, which leads to the following system of differential equations for the adjoint prototypes,

$$\begin{aligned}\dot{v}_k^+ &= \lambda_k(v_k^+q)q - b\sum_{l\neq k}^{n}(v_k^+q)(v_l^+q)^2q - c\sum_{l=1}^{n}(v_k^+q)(v_l^+q)^2q + \\ &\gamma[2(v_k^+q)q - \sum_{l=1}^{n}(v_k^Tv_l)(v_l^+q)q - \sum_{l=1}^{n}(v_k^Tq)(v_l^Tq)v_l^+].\end{aligned} \tag{5.6}$$

The prototypes are learnt simultaneously, and their evolution is described by,

$$\dot{v}_k = \gamma[2(v_k^+q)q - \sum_{l=1}^{n}(v_l^+q)(v_k^+q)v_l - \sum_{l=1}^{n}(v_k^+v_l^{+T})(v_l^Tq)q]. \tag{5.7}$$

There are a number of penalties which must be paid for introducing the extra potential term. First, we need to adjust γ so as to weight the second potential term carefully. A poor choice of weighting will mean that we are not truly minimising the synergetic potential, $\hat{p}$. Second, the resulting dynamical system is considerably more complex. Third, the extended potential requires that we calculate the prototypes and adjoint prototypes simultaneously, thereby doubling the number of equations to integrate. If we wish to learn four classes of images, each of which has just 100^2 pixels, this amounts to integrating a system of 80,000 equations. While it is envisaged that this will be practical for the construction of synergetic hardware [102], software implementations are unwieldy. Fourth, and most important, Haken has shown that the original synergetic potential is a Lyapunov function for the learning process [40, p. 107]. With the addition of extra terms, no such proof exists, and so the absence of false minima can no longer be guaranteed.

We need one further element before this system can be used for unsupervised learning. In the case when we have n training images and n classes, the system should define each training image as a prototype and the learned connectivity matrix should be identical to the one which would have been produced using SCAP. This will be the case if the potential finds its global minimum during the learning process. However, we have no guarantee that distinct training images will start in different basins of attraction, and if they begin in the same basin, they will converge to a single final memory. Haken [40] has found a necessary and sufficient condition which precludes this situation. If the initial conditions satisfy,

$$|(v_k^+q_k)| > |(v_k^+q_l)| \qquad \forall l \neq k, \quad k = 1, \ldots, m, \tag{5.8}$$

then the unsupervised learning will emulate the supervised learning as desired.

Unfortunately, these initial conditions which ensure that the system reaches a global minimum are problematic, as they pre-suppose a knowledge of the required answer.

5.4 Enhanced Unsupervised Learning

There are four major problems with the approach just described. The complicated method of constraining the answer, the need to tailor the initial conditions, the sheer size of the equation set that needs to be integrated and the unknown effect of the extra terms on the potential. We remove all four of these problems with one observation.

A vector $\boldsymbol{v}_k^+$ lies in the space spanned by the set of vectors $\boldsymbol{q}_j$, if it can be expressed as a linear superposition of the $\boldsymbol{q}_j$. Thus if we construct the adjoint prototypes as a linear superposition of the $\boldsymbol{q}$ vectors, our requirement will automatically be met without the need for an extra potential term.

Poulton [86] has interpreted the task of finding the adjoint prototypes as a Lagrange multiplier optimisation problem and shown that they are in fact, a linear superposition of the prototypes transposed. Our restriction on the subspace of the adjoint prototypes is therefore met if the prototypes are a linear superposition of the training images. This situation is described by,

$$V^+ = AV^T, \qquad V = QG^T, \tag{5.9}$$

where A is a square superposition matrix of side-length n, G is the superposition matrix of size $(m \times n)$, Q is the matrix of $\boldsymbol{q}_j$ vectors and the superscript T represents the transpose operation.

We can calculate A by substituting Equation (5.9), into the orthonormality requirement, $V^+V = I$,

$$A = (V^TV)^{-1} = (GQ^TQG^T)^{-1} \tag{5.10}$$

We now define the *order parameters*, $\xi = V^+Q$ and use Equation (5.9) to give,

$$\xi = AGQ^TQ. \tag{5.11}$$

Now the potential p can be expressed in terms of the order parameters as,

$$p(V^+, \boldsymbol{q}_j) = -\frac{1}{2}\sum_{k=1}^{n}\lambda_k\xi_{jk}^2 + \frac{b+c}{4}\sum_{k=1}^{n}\sum_{l=1}^{n}\xi_{jk}^2\xi_{jl}^2 - \frac{b}{4}\sum_{k=1}^{n}\xi_{jk}^4. \tag{5.12}$$

Applying Equation (5.3) gives a potential $\hat{p}(G, Q)$, which is a function of the $(m \times n)$ elements in the G matrix. For the previously described situation, we have reduced the number of variables from $80,000$ to four times the number of training images.

Now by minimising the potential we will calculate an optimal G matrix which defines our final set of learned prototypes. Due to the massive reduction in the number

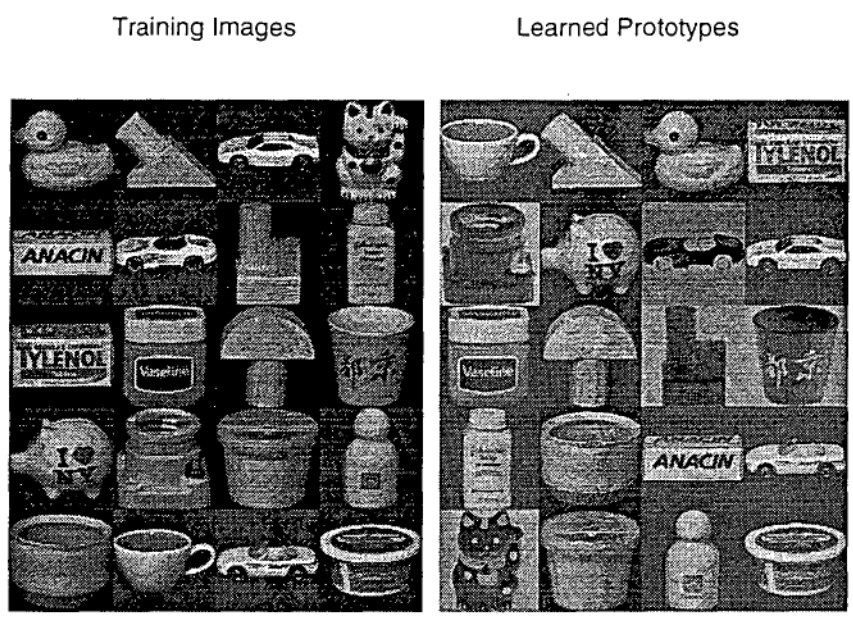


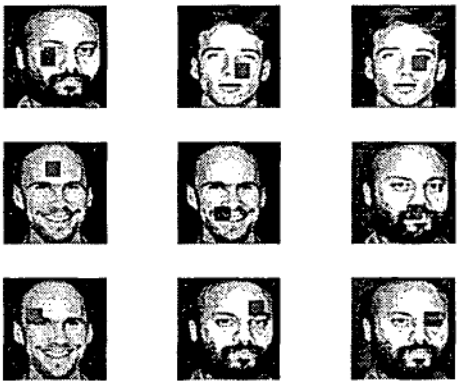
Figure 5.2: Training images and learned prototypes

of variables, we can use a sophisticated minimisation routine, namely the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method with a mixed quadratic and cubic line search procedure [87]. This procedure is far less likely to find a local minimum than gradient descent, and so the need for using tailored initial conditions is removed. If the system were to find a local minimum, this would be clearly visible in the final potential returned by the minimisation routine, in which case, a new random initial condition could be used until the desired result was found.

At each iteration, G is scaled such that the resulting prototypes have unit length and A is calculated using Equation (5.10). The initial $G(0)$ matrix may be chosen randomly.

5.4.1 Unsupervised Supervised Learning

One of the challenges of unsupervised learning is how to measure the success of a learning process. In general, there is no one correct result, only a sense of what might be considered reasonable. One test in which the reasonable answer is obvious is that of reproducing supervised learning of n training images to produce n prototypes. To test this we used a dataset of twenty objects from the COIL database [76]. The training images and the learned prototypes are shown in Figure 5.2, where the different grey-levels are due to automatic scaling within the graphics program. Note the inversion of a number of the prototypes, which is allowed in the formalism of synergetics [41]. It is also worth noting that this result has been achieved without the need to construct favourable initial conditions as required by the standard approach [40].



(a) Subset of the training images.

(b) Learned Prototypes

Figure 5.3: Learning prototypes from a set of occluded images.

5.4.2 Learning from a Noisy Training Set

Our unsupervised learning system is also capable of learning prototypes from a set of corrupted training images.

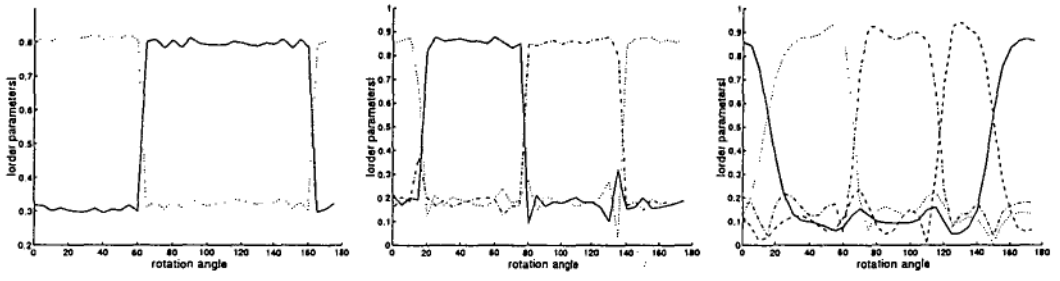
Given images of three faces we created a database of 60 training images, each of which was a face occluded by uniformly grey squares. Figure 5.3 shows a selection of training images and the resulting prototypes. Note that in this case, the system has actually *reconstructed* each complete face.

5.4.3 Learning a Concept

Just as in supervised learning, the idea of unsupervised learning is to learn a *concept*. The distinction is that the system has to form its own concept without guidance from a teacher. There may, however, be several reasonable concepts within the same dataset. A training set of facial images could, for example be broken into male and female, or into those that do and do not wear spectacles.

We have chosen to test the system on learning the concept of pose. We follow Wagner et al. [102] and introduce a new training database containing images of a single, wide line segment rotated in the plane at an angle around the centre. There are 36 images, covering 0° to 175° in 5° steps, a sample of which are shown in Figure 5.4. In Figure 5.5 the vertical axis gives the absolute value for the order parameters, which are the synergetic measure of similarity, and the horizontal axis gives the angle of rotation. Clearly, the system has successfully learned the concept of pose. It has

Figure 5.4: Examples of images from the database of rotated lines.



(a) 2 classes (b) 3 classes (c) 4 classes

Figure 5.5: The largest order parameter determines the classification.

divided the continuous set of training images into 2, 3 and 4 classes, based on their rotation angle.

5.5 Conclusions

Synergetic learning is a recent concept which treats pattern learning as a type of pattern formation. This idea is quite intuitive, matching well with the accepted example-based learning of neural networks, which is itself loosely modelled on neurological behaviour.

Importantly, the pattern formation model used for learning is exactly the same as that which has been used for recognition in previous chapters. This confluence of what have often been treated as two separate activities in engineering, is in contrast to many neural networks in which there are distinct learning and recognition phases.

Our implementation of pattern formation as learning is a major improvement over the original formulation. It dramatically reduces the dimension of the system, as well as simplifying the individual elements in the system. It is also robust because it is based on minimisation of a Lyapunov function. These improvements over the previous algorithm are made possible by understanding and explicitly enforcing the relationships between the training set and the prototypes. As a result of these changes, our implementation is a practical approach for a software based unsupervised learning system.

The decision to explicitly enforce the prototype restrictions, however, has more implications. We can now envisage similarly practical algorithms for two related areas. First is the challenge of *unsupervised updating*, where a memory, which has already been trained, either in a supervised or unsupervised fashion, is to be extended to

incorporate new patterns. Second is the idea that as both recognition and learning can be carried out by the same mechanism separately, we can also devise systems in which learning and recognition happen *simultaneously*. Mathematically, we simply make both the prototypes and training images evolve in time by the addition of linear combinations of the patterns. These evolutions are controlled by a user defined scalar which defines the ratio of learning to recognition. The final result is that which, via the evolving order parameters, minimises the synergetic pattern formation potential.

Another interesting area for investigation is that of combining a synergetic pattern formation/rejection potential, such as the one introduced in Chapter 4, with an unsupervised updating routine. Such an approach could result in a system which rejected patterns that did not belong to the classes in memory, but which was capable of extending the concept of each class by capturing new information in a novel pattern.

Part B

Synergetic Pose Estimation

Chapter 6

View-Based Pose Estimation

Pose estimation is the task of estimating the pose, or rotation parameters, of an object. The literature on this topic is relatively small, but it is growing quickly in line with the increase in inexpensive computational power which makes such algorithms practical, and the increasing numbers of applications, such as autonomous navigating robots, which motivate the research.

We have illustrated the large variety of differing approaches by Figure 6.1.

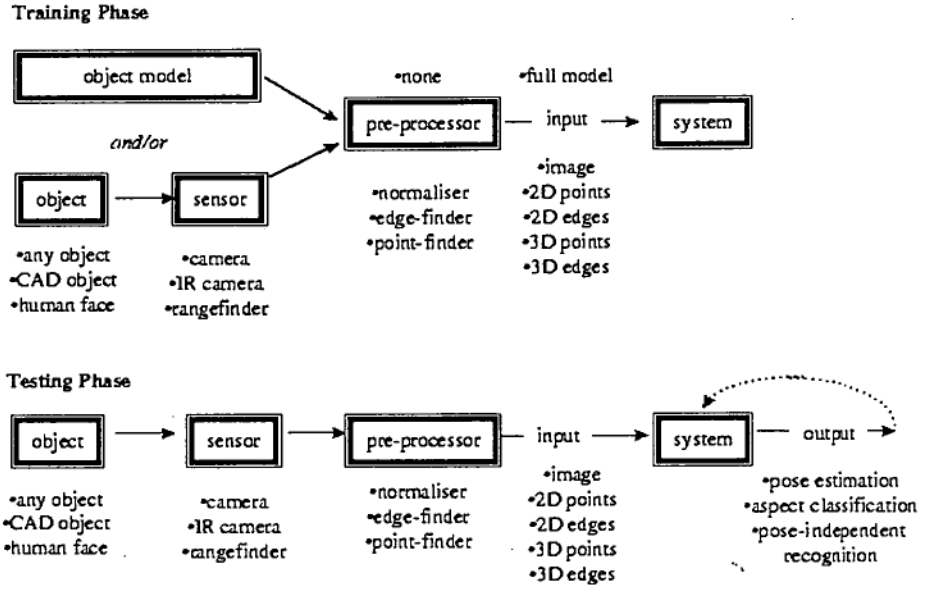


Figure 6.1: Schematic diagrams for generic pose estimation system.

Amongst these various possibilities there are four major distinguishing factors.

- First, if the training phase requires a full CAD model of an object, we refer to it as a *model-based* approach to pose estimation. Alternatively, a system which is trained using a number of examples of the object rotated at different poses is

referred to as an *example-based* estimator.

- Example-based estimators can be subdivided further by the second consideration, which is the type of sensor used. If, like a standard video camera, the sensor captures a two-dimensional representation of the three dimensional object, we refer to the system as *view-based*. In contrast, a sensor which captures three-dimensional information gives a *range-based* system.

- The third important element in classifying the available algorithms is their purpose. Some algorithms return a quantitative estimate of the test object's pose, and are labelled *pose estimation* systems. Others return an *aspect*, which was originally defined as a topologically equivalent class of object appearances [58], but which can be thought of more intuitively as a contiguous subdomain of pose space in which the object looks qualitatively the same. Thus the result is a qualitative algorithm or *pose classifier*. Finally, a number of algorithms attempt to achieve *pose independent object recognition*, and calculate an estimate of the pose as an intermediate step.

- The fourth major classification issue is the genericy of the approach. For certain important problems, such as estimating the pose of a human face, application-specific approaches have been built. We label these object-specific as opposed to object-independent.

6.1 Review

6.1.1 View-Based Pose Estimation

In 1979 Ullman [100] reported an important result on pose estimation from images. He found that the pose of a rigid object could be found uniquely (up to a reflection), using three orthographically projected images of the object with four corresponding points on each image. There are a number of related results [52, 49, 80], but the problem of finding related points, the so-called *correspondence problem* is a difficult one which limits the practical use of these findings.

A different approach is to use global features, such as *image moments*, which are a popular form of feature for image processing [88], and more specifically, for estimating the pose of an object [22, 89]. The moment functions of the image intensity distribution are defined as surface quadratures and are used to represent the global shape characteristics of the image.

Mukundan [69] used a quaternion representation of the pose parameters and found a closed form solution using geometric moments of up to order two. Rotational ambiguities were resolved using non-moment information from the image. Mukundan et al. [70] later added the use of moment invariants for calculating the camera-view axis. Moment invariants are functions of moments that are independent of scale and in-plane rotation. Later still, they generalised their approach to allow a full affine transformation of the object, encompassing translation, rotation and scale. The solution of this system required the use of the third order moments and was found iteratively [71].

Brochard et al [14] have extended the moment approach by applying it to five separate camera images of the same object. They establish a new model of rigid-body movement in three-dimensional space based on the bi-dimensional moments of an object's orthogonal projections, given the assumption that the object has a matt, convex surface and uniform lighting.

An alternative global feature set which captures the shape of the object is the Fourier descriptor [89] as used, for example, by Man et al. [64]. When an object, defined by a series of points from the outline of the object, is projected onto the plane of the monitor, the authors show a relationship between the Fourier descriptors of the planar object and those of the projected object. The solutions to these equations then yield approximations for the slant and tilt angles of the original object with good robustness to randomly distributed noise.

Fairney and Fairney [24] have argued that these global measures of shape are susceptible to occlusions and other local noise structures. For this reason, they use a local measure of shape, namely linear boundary segments from the object's silhouette. They store a training set of such feature sets and use a pose clustering algorithm to find the closest match. As a result of their local shape measures, the resulting pose estimator is demonstrated to be highly robust to occlusions and localised noise.

Ikeuchi and his co-workers [54] group appearances of simple objects obtained by tesselating the viewing sphere, into aspects. This can be done with use of a CAD model or from a planned set of observations of the object. They define an interpretation tree as the basis of a recognition strategy which classifies a novel image as belonging to a specific aspect. A second system is then used to estimate the pose within that aspect.

Poggio et al [84, 83, 82] have developed a general purpose image analysis network based on generalised radial basis functions. It is general purpose in that it can be used to estimate parameters other than pose. For example, facial expression was analysed in the more recent reference. They have also extended their approach to produce a parallel synthesis network, which constructs an image from a canonical image and a series of transformations. Both the analysis and synthesis networks require the calculation of pixelwise correspondences, which is a difficult and computationally expensive problem.

Murase et al [74, 75] introduced a *parametric eigenspace representation* of an object to produce a general purpose image analysis algorithm. Given an object, they produce a low dimensional hypersurface representation, which is parameterised by the variables which are to be estimated. In this case, they use one pose variable and one lighting direction variable. Since images of a single object will be highly correlated they used Principal Component Analysis to reduce the dimensionality of the representation by finding the most significant eigenvectors of the training set images. These eigenvectors span a subspace known as the eigenspace, into which the test images were projected to form the parametric representation. Once a suitable representation has been found for an image, a novel image is analysed by projecting it into the eigenspace and finding the closest point on the representation surface. The novel image is assigned the parameters belonging to that surface point.

For particular objects of interest, object-specific approaches can use knowledge of the object or restrictions specific to a given application. A good example of this is

for very low bit rate teleconferencing applications, in which a system is required to estimate the pose of a human face so that this information can be transmitted and the face reconstructed. Brunelli [15] restricts the problem to the rotation found in a shaking gesture, and measures the asymmetry of the two eyes as the basis for a pose estimate. Tsukamoto et al. [96] allow for variation in all three rotation angles but take advantage of the restricted range of the variables to estimate pose. Beymer [9] uses the structure of the human face by finding the eyes and nose of the subject, before applying a template matching procedure to estimate the pose.

6.1.2 View-Based Pose-Independent Object Recognition

Most computer vision systems perform object recognition based on features extracted from a single sensed instance of the object. While in many practical situations this may be necessary, it does make the implicit assumption that any one instance of the image holds enough information for this to be possible. Common experience, however, assures us that this is often not the case, as illustrated by the back of a human head.

Gremban and Ikeuchi [34] introduced a view-based object recognition system called a *vision algorithm compiler* or VAC. A VAC analyses object images and creates appropriate object recognition code off-line. The recognition procedure is then executed for a novel image in the on-line phase. They have also addressed the question of which images should be used as a training set for a VAC [35].

The Murase [74, 75] scheme for pose estimation can also be used effectively for pose independent object recognition. A parametric representation is found for each possible object and a novel image is projected into each of the eigenspaces. The object and its pose can be estimated simultaneously by finding the closest surface point across all of the representations.

Black and Jepson [10] used the same eigenspace for finding a given object in an image and tracking its position throughout a video sequence. The tracking uses optical flow and is designed to be robust to changes in pose as well as motion in articulated objects. The task of tracking an object uses the dashed arrow shown in Figure 6.1 where the output from the previous time-step is used as a starting point for the system in the current time-step.

Seibert et al. [92] also use a view-based approach to object recognition over a period of time. They, however, use the time information to continuously re-assess their classification hypothesis. In a similar manner to Fairney and Fairney [24], they cluster images into aspects, but they extend this to analyse transitions between aspects. This process allows the changes in image pose to be used to distinguish between possible object classifications.

6.2 Uniqueness, Equality and Ambiguity

The three rotation angles around the x, y and z axes are an obvious choice for representing pose. Unfortunately they are not *unique*. It is possible to describe the same rotation with two different angle sets because rotations are not commutative. Thus

we can find a set of angles, a, b, c and d, where $a \neq d$ and $b \neq c$, and yet

$$R_x(a)R_y(b) = R_y(c)R_x(d). \tag{6.1}$$

In order to remove confusion resulting from this, we enforce the rotation order $x \rightarrow y \rightarrow z$.

Even having enforced an order of rotation, and normalised angles to a standard interval, say $0 \leq a < 2\pi$, there are always at least two ways of representing the same pose. If we select an arbitrary set of angles, a, b and c, then

$$R_x(a)R_y(b)R_z(c) = R_x(\pi + a)R_y(\pi - b)R_z(\pi + c), \tag{6.2}$$

is an identity [33].

Depending on the shape of an object, there are also instances of *view equality*. This occurs when two different poses produce the same image, and is particularly common in objects with symmetry. For example the sphere, which is symmetric around all axes passing through its centre is a degenerate object in terms of pose, because every possible orientation produces the same image. Clearly, view equality is a problem for any pose estimation system. The most common view equalities occur with rotations of 180°, but they can occur elsewhere. In practical terms *view ambiguity*, which is defined roughly as instances where the views are similar, is equally problematic.

Clearly, a situation of view equality must yield equal feature sets from the images, leading to *feature equality*. Similarly, we would expect that view ambiguity would lead to *feature ambiguity* or *actual ambiguity*. It would probably be possible to extract a distinct, object-specific feature from the images such that the feature sets would not be ambiguous, but the genericy of the approach would be diminished. In choosing a feature extractor, our aim is that no two images which do not suffer from view ambiguity will produce feature sets that have feature ambiguity. When this aim is not met, the resulting feature ambiguity is *artefact ambiguity*.

The first approach to dealing with ambiguity is for the system to return all possible values. For certain applications, this would be sufficient.

In the case of feature equality, the only way to differentiate the alternatives is to use extra information, not present in the images. For example, if one is tracking the pose of a single object over time, the previous pose estimate could be used to decide between the alternative pose values.

Instances of artefact ambiguity suggest a poor choice of feature extractor. Rather than attempt to resolve the ambiguity, the best approach would be change or supplement the features such that the artefact ambiguity is resolved.

Unlike equality, ambiguity is not a precise concept, so we can find different levels of ambiguity in different situations. When the ambiguity level is high enough, the difference between the features will be corrupted by the noise, and so we proceed as we would for feature equality. For lower levels of ambiguity, *pose space subdivision*, which is described in Section 8.5.1, can help to distinguish between the alternatives.

6.3 Conclusions

View-based pose estimation is a relatively new approach to pose estimation, which does offer a viable alternative to traditional CAD-model based approaches under cer-

tain conditions. The applicability of view-based algorithms depends on the particular situation involved. In general they perform well in situations where the viewing conditions are consistent, such as those found in manufacturing environments. Their main strength is that they can be applied to any object, as images at known pose are inexpensive to produce.

Their major weakness is a lack of robustness to changes in viewing conditions. The appearance of an object changes, for example, with lighting and camera location variation, whereas the definition of a CAD model remains constant. The solution to this problem must come from a more sophisticated approach to extracting features from the image to ensure, for example, that there is no artefact ambiguity.

A major challenge facing both model-based and view-based pose estimation systems is speed. In both of the approaches, it is necessary to search feature-space for a closest match to the test input. This requirement is problematic when the application calls for tracking the pose of an object in real-time.

Chapter 7

Synergetic Warping

7.1 Motivation

In Part A we discussed a simple model of pattern formation and showed how it could be used to implement a practical synergetic pattern recognition scheme. This is a qualitative classification scheme. In this and the following chapters, we investigate whether the same concept can be quantified and used to investigate patterns which are controlled by continuous parameters, such as pose. As a result of this investigation, we have developed two new methods for synergetic pose estimation.

Our own experience states that there is sufficient information in an image of a familiar object, to be able to estimate its pose. While the question of how we do this is still open, a significant volume of research [93, 48] suggests that we store a number of *canonical views* [105] of an object, and interpolate between them by performing a *mental rotation.* This theory is also supported by the commonly experienced phenomenon that it can be difficult to recognise an object which is far from a canonical rotation, such as when it is upside down.

Fortunately, some of the framework required to implement this principle within a synergetic algorithm has already been put in place by Haken and his co-workers [40, 16]. They introduced a method of transforming images as a pre-processing step before carrying out standard synergetic recognition. The method described in this chapter also applies a transformation to images and uses standard synergetic recognition, but it is different in a number of important ways. First, the transformation is an out of plane rotation, in contrast to the in-plane transformations used previously. Second, we have implemented the transformation and recognition sections to occur concurrently.

7.2 Concept

Synergetic warping, our first method of synergetic pose estimation, applies the principles of canonical views and mental rotation within the construct of synergetic pattern recognition. Synergetic warping uses the standard synergetic potential, but is distinguished from the standard recognition process in several ways.

- Each canonical view is stored as a prototype, and the final state of the system will be close to, but not in general, equal to, one of the prototypes.

- The evolving image is a transformation of the original image, not a linear superposition of the prototype images.
- The transformations are restricted to the set of rotations viewed with perspective, thereby implementing the concept of a mental rotation, and
- the evolution is measured *quantitatively.*

The metric used here is classified by Basri et al. [6] as a *transformation metric* because it measures the deformations applied to an object to produce an observed image. In contrast, most pose estimation systems use an *image metric* which measures the distance between two images of the object.

7.2.1 Synergetic Warping Potential

We start by re-expressing the standard synergetic potential as the synergetic warping potential,

$$\begin{aligned} p_{warp} &= -\frac{1}{2}\sum_{k=1}^{n}\lambda_k(\boldsymbol{v}_k^+\boldsymbol{q}(\beta,f))^2 \\ &\quad +\frac{1}{4}\sum_{l\neq k}\sum_{k\neq l}B_{kl}(\boldsymbol{v}_l^+\boldsymbol{q}(\beta,f))^2(\boldsymbol{v}_k^+\boldsymbol{q}(\beta,f))^2 \\ &\quad +\frac{1}{4}c\sum_{kl}(\boldsymbol{v}_l^+\boldsymbol{q}(\beta,f))^2(\boldsymbol{v}_k^+\boldsymbol{q}(\beta,f))^2, \end{aligned} \tag{7.1}$$

where the evolving image $\boldsymbol{q}$ is a function of the rotation angles β and the focal distance of the perspective system, f, both of which are varied so as to minimise the potential.

Apart from the dependence on β and f, this is identical to the standard potential and the three terms of the potential function play their familiar roles. The first term deforms the test image $\boldsymbol{q}$ towards one of the prototypes. The second term introduces competition between the prototypes and the third term enforces the normalisation of the image.

7.2.2 Perspective Rotation Transformation

Our goal in transforming the image $\boldsymbol{q}$ is to mimic as closely as possible, the image which we would have seen, had the given object been rotated by β degrees from its current pose and viewed with the focal length f. We therefore define the transformations represented by the function $\boldsymbol{q}(\beta, f)$ to achieve this goal.

The given image is a 2-dimensional projection of a 3-dimensional object onto the plane of the monitor. Rather than change the greyscale values of the image directly, we place the image on a square grid defined by the coordinates $\boldsymbol{u} = (u_x, u_y, 0)$. Now we rotate the square grid around the x, y and z axes before projecting the warped grid back onto the plane of the monitor using the given focal length, f. The resulting warped grid defines our transformed coordinates, $\tilde{\boldsymbol{u}} = (\tilde{u}_x, \tilde{u}_y, 0)$.

Using homogeneous coordinates for the sake of mathematical simplicity ([104]), the desired transformation can be written as

$$\tilde{u} = uRP, \tag{7.2}$$

where P and R are matrices that represent the projection and rotation operators respectively. Since R consists of three rotations around the x, y and z axes, we can write $R = R_x R_y R_z$ in homogeneous coordinates, as

$$R_x(\beta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\beta_x & sin\beta_x & 0 \\ 0 & -sin\beta_x & cos\beta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.3}$$

$$R_y(\beta_y) = \begin{bmatrix} cos\beta_y & 0 & -sin\beta_y & 0 \\ 0 & 1 & 0 & 0 \\ sin\beta_y & 0 & cos\beta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.4}$$

$$R_z(\beta_z) = \begin{bmatrix} cos\beta_z & -sin\beta_z & 0 & 0 \\ sin\beta_z & cos\beta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7.5}$$

The projection operator introduces the perspective effects,

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & f \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{7.6}$$

where f is the focal length of the system. This projects the point (x, y, z) onto $(\frac{x}{fz+1}, \frac{y}{fz+1}, 0)$.

Expanding Equation 7.2 yields two equations,

$$\begin{aligned} \tilde{u}_x &= \frac{au_x + bu_y}{rcu_y + rd_x + 1} \\ \tilde{u}_y &= \frac{eu_x + gu_y}{rcu_y + rd_x + 1}, \end{aligned} \tag{7.7}$$

where

$$\begin{aligned} a &= cos(\beta_2)cos(\beta_3) \\ b &= sin(\beta_1)sin(\beta_2)cos(\beta_3) - cos(\beta_1)sin(\beta_3) \\ c &= sin(\beta_1)cos(\beta_2) \\ d &= -sin(\beta_2) \\ e &= cos(\beta_2)sin(\beta_3) \\ g &= sin(\beta_1)sin(\beta_2)sin(\beta_3) + cos(\beta_1)cos(\beta_3). \end{aligned}$$

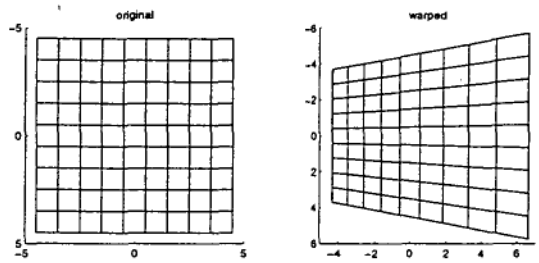


Figure 7.1: The original and transformed grids.

Now to find the required transformation, we solve Equation 7.7 for u_x and u_y, which gives,

$$u_x = -\frac{\tilde{u}_x(e\tilde{u}_x - a\tilde{u}_y)}{(e\tilde{u}_x - a\tilde{u}_y)(rc\tilde{u}_x - b) + (b\tilde{u}_y - g\tilde{u}_x)(rd\tilde{u}_x - a)} \tag{7.8}$$
$$u_y = -\frac{\tilde{u}_x(b\tilde{u}_y - g\tilde{u}_x)}{(e\tilde{u}_x - a\tilde{u}_y)(rc\tilde{u}_x - b) + (b\tilde{u}_y - g\tilde{u}_x)(rd\tilde{u}_x - a)}.$$

The warping of the image grid is shown in Figure 7.1, which shows both the square grid and the transformed coordinate system derived by applying the R and P transformations.

Having calculated the transformed coordinate system, we now warp the image by a two-dimensional linear interpolation process, such that the greyscale value at $\tilde{\boldsymbol{u}}$ in the warped image is equal to the greyscale value at the equivalent point $\boldsymbol{u}$ in the original image. Pixel values that are not defined by this transformation are set equal to the background in the original image. Figure 7.2 shows the three stages in the image transformation. The first image is the original, the second image has been transformed but has undefined pixel values in the left hand corners, and the third image shows the final image where these pixels have been made part of the background.

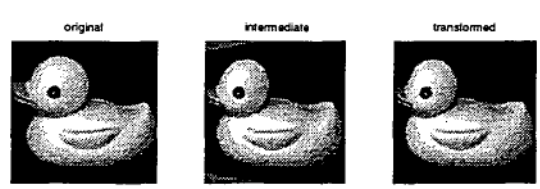


Figure 7.2: The original, intermediate and transformed images.

It is important to emphasise the separation between the synergetic warping potential and the particular transformation described above, because the concept of minimising the synergetic warping potential can be applied using any transformation.

The given transformation is sufficient to demonstrate this concept, yet it is naive in a number of ways. First it has no knowledge of the shape of the object. Second, it assumes that the axis of rotation for the object is in the centre of the image. Third, it only uses a single image, ignoring the object information available in the prototype images. While no transformation based on two-dimensional images will be able to perfectly mimic the effect of a rotation of a three-dimensional object, all three of these assumptions could be removed or restricted to form the basis of a more sophisticated transformation and hence a more robust pose estimation system.

The major source of error introduced by this transformation is the assumption that all of the points in the image lie on a plane parallel to the monitor. When the object is rotated in such a way as to make the validity of this assumption poor, the accuracy of the system will decrease significantly, as will be illustrated by the examples.

7.3 Pose Estimation with Synergetic Warping

As a view-based pose estimation routine, synergetic warping needs access to a training set of images which are correctly labelled with the pose values used to create the image. Let each one of these n images be a prototype, $\boldsymbol{v}_k$, and let $\boldsymbol{\alpha}_k$ be a vector defining the pose of the kth prototype. In the standard way, we normalise the prototypes and construct the adjoint prototypes using Equation 2.18.

Given a test image $\boldsymbol{q}$ with unknown pose, we wish to find a $\hat{\boldsymbol{\beta}}$ and an $\hat{f}$, which will warp $\boldsymbol{q}$ to look as close to one of the prototypes as possible. They will not be identical, as is the case in pattern recognition, because the transformations are constrained to perspective rotations.

We could attempt to minimise the potential using the familiar gradient descent approach, but the relationship between the order parameters and the transformation parameters is non-linear and the likelihood of local minima is high. Considering that we have just four variables, we choose to use again the Broyden-Fletcher-Goldfarb-Shanno minimisation routine [87] to find $\hat{\boldsymbol{\beta}}$ and $\hat{f}$. Fortunately, the choice for the initial value of $\boldsymbol{\beta}(0)$ is clearly the zero vector. For $\hat{f}(0)$, we use a focal length typical of the focal lengths in the training set.

When the minimisation routine has converged, the outputs are $\hat{\beta}$, $\hat{f}$ and i, the index of the prototype closest to the warped image. We then calculate an estimate of the novel pose, $\hat{\alpha}$, as

$$\hat{\alpha} = \alpha_i - \hat{\beta}. \tag{7.9}$$

Figure 7.3 illustrates this process in the case of the rubber duck shown in Figure 7.2, which is rotated around its natural axis of rotation by an angle, β. In this instance the minimisation ended with a value of $\beta = 8°$ and the largest order parameter belonged to the prototype image rotated at 30°. Since the new image was rotated 8° to match the prototype as closely as possible, the estimated pose for the test image is given by $30 - 8 = 22°$.

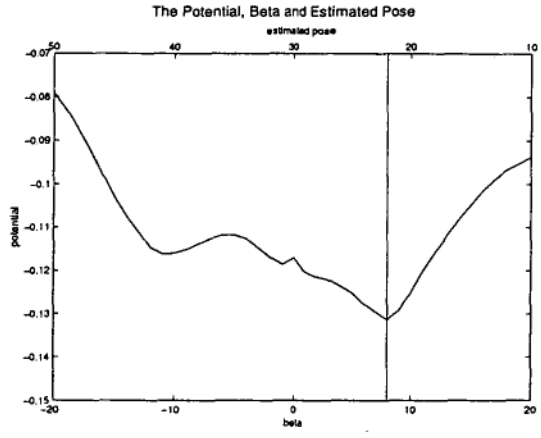


Figure 7.3: The minimum of the potential gives a value for beta and an estimate of the pose.

7.3.1 Synergetic Prototype Warping

In our implementation we have transformed the test image such that it resembles the prototypes. This is in contrast to Haken et al. [40, 16], who deformed each prototype in parallel to resemble the test image.

This distinction suggests an alternative approach to pose estimation using synergetic warping based on transforming the prototypes. We will call this technique *synergetic prototype warping*, and the appropriate synergetic potential is given by,

$$\begin{aligned} p_{SPW} &= -\frac{1}{2}\sum_{k=1}^{n}\lambda_k(\boldsymbol{v}_k^+(\boldsymbol{\beta},f)\boldsymbol{q})^2 \\ &\quad +\frac{1}{4}\sum_{l\neq k}\sum_{k\neq l}B_{kl}(\boldsymbol{v}_l^+(\boldsymbol{\beta},f)\boldsymbol{q})^2(\boldsymbol{v}_k^+(\boldsymbol{\beta},f)\boldsymbol{q})^2 \\ &\quad +\frac{1}{4}c\sum_{kl}(\boldsymbol{v}_l^+(\boldsymbol{\beta},f)\boldsymbol{q})^2(\boldsymbol{v}_k^+(\boldsymbol{\beta},f)\boldsymbol{q})^2, \end{aligned} \tag{7.10}$$

where the rotation vector $\boldsymbol{\beta}$, now holds the rotation angles for all n prototypes.

In choosing between the two synergetic warping approaches, we must compare accuracy with complexity and speed. There are two reasons why synergetic prototype warping is potentially more accurate.

First, the fact that standard synergetic warping assumes that the object is two-dimensional, is likely to be a source of significant errors. In order to calculate the deformation due to the rotation, it is necessary to assume that all elements in the image lie in the same plane $z = z_0$. As we have no real knowledge of the test image, this is the most reasonable assumption available to us.

However, if we deform the prototypes, we can use a more sophisticated assumption because we know more about the prototypes. A simple possibility would be to continue the assumption that the object is planar, but loosen the assumption that the plane be

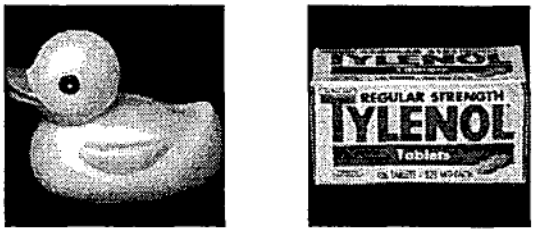


Figure 7.4: Two objects on which synergetic warping for pose estimation was tested.

parallel to the plane defined by the x and y axes. In the case that an object has an axis significantly larger than the other two axes, this assumption should significantly increase the accuracy with which the deformation of the image mimics the rotation of the object.

A second possibility is to use *shape from shading* algorithms on each of the prototypes. These techniques attempt to estimate the distance from the camera to each point in an image within an additive constant, based on assumptions about the reflective properties of the object. The most common assumption is that the object has a Lambertian surface. Having estimated the associated z values as part of the pre-processing, we can then use this extra information as the starting point for the deformation process.

The second reason why synergetic prototype warping is likely to be more accurate is that if there is significant noise in the test image, standard synergetic warping is likely to exacerbate the effect in the deformation. In contrast, we can pre-process the prototype images to remove as much noise as possible to minimise this effect.

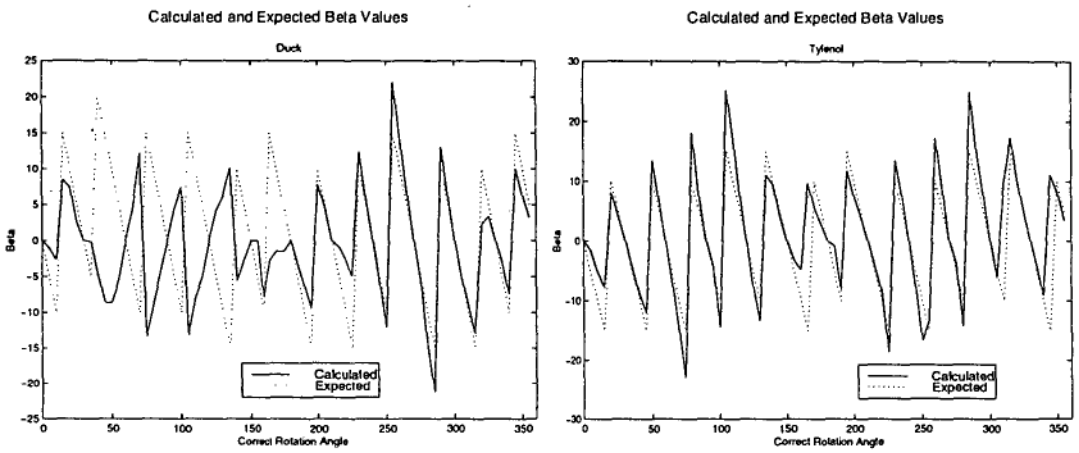
Unfortunately, there is a considerable increase in the complexity required for synergetic prototype warping. First, in standard synergetic warping we have a maximum of four variables to be ascertained: the three rotation parameters and the focal length. In contrast, n prototypes require a maximum of $3 \times (n + 1)$ variables for synergetic prototype warping. Second, we must deform the prototypes but the potential of Equation (7.10) is dependent on the associated adjoint prototypes, so we must calculate these at each time step. Calculating these involves inverting an $n \times n$ matrix, as shown in Equation (2.18) so this requirement is clearly problematic.

7.4 Examples

To test the concept of synergetic warping for pose estimation we have estimated the pose of two objects as shown in Figure 7.4 from the COIL database [77].

Each image is rotated in a complete revolution around a natural axis in 5° steps. Empirical evidence [90] suggests that humans have difficulty in recognising or imagining wire-frame objects in a novel orientation that differs by more than 30° from a known view, so we selected images in 30° steps as our prototypes. We do not know the focal length used to capture images in the COIL database, so we set an arbitrary but reasonable focal length of 100 pixels.

The results for the two different objects were markedly different. This is reflected



(a) Duck. Note the opposite signs from approximately 45° to 145°.

(b) Tylenol.

Figure 7.5: The calculated and expected values of β for each object.

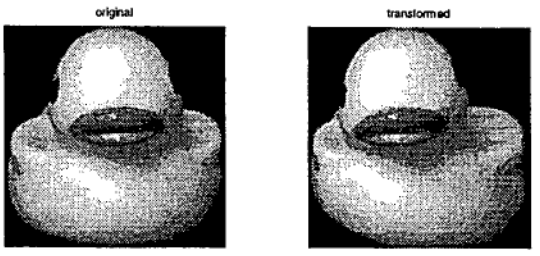
in the errors listed in Table 7.1 but can be seen most prominently in the comparison of the two parts in Figure 7.5. The test set consisted of the 60 images not in the training set.

Errors	Mean	Max
Duck	8.18	28.4
Tylenol	3.83	24

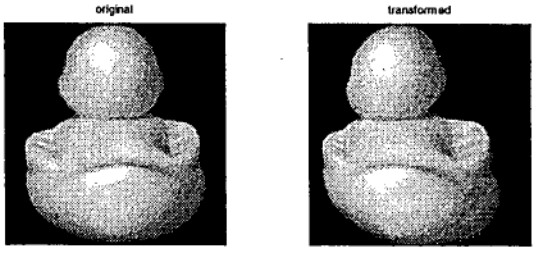
Table 7.1: Errors using synergetic warping.

Note in Figure 7.5(a) that when the duck is rotated between approximately 45° and 145°, the sign of the calculated value of β is incorrect, but the magnitude is close to the expected value. This behaviour is due to the combination of a simplistic rotation transformation and a relatively complicated shape. To be more specific, when the duck is rotated at 90°, which is at the centre of the problematic domain, it is facing directly away from the camera. Now because the duck's head is at the furthest distance possible from the camera and its tail is as close as possible, the assumption that the object lies on a plane parallel to the monitor is a very poor one.

While this argument explains the breakdown of the pose estimation system, we must investigate the symmetry of the object to understand why the results are incorrect by a factor of -1. When the head of the duck is facing the camera, a clockwise rotation would move the duck's head to the left of the image. This rotation has been successfully approximated by the image transformation, as can be seen in Figure 7.6(a). When the duck's head is distant from the camera, the same rotation should see the head move to the right, but because the planar assumption puts the head on a plane, essentially at



(a) The head of the front-facing duck image moves correctly to the left.



(b) The head of the rear-facing duck image moves incorrectly to the left.

Figure 7.6: Duck images transformed to mimic a rotation of 20°.

the front of the image, the head is seen to move in the wrong direction (Figure 7.6(b)). Because the image of the duck is symmetric when rotated to 90°, we therefore find that the calculated values for β have approximately the expected value but the wrong sign.

In contrast, the Tylenol package is a simple object without outstanding features, such as the duck's head. The improved accuracy found using this object over the duck can be seen quantitatively in Table 7.1. This can also be seen qualitatively in Figure 7.5(b) which shows a good correlation between the calculated and expected β values across the entire revolution.

7.5 Conclusions

Synergetic warping for pose estimation is of interest for two reasons. First, it is synergetic in the strict sense of the word, because it is based on competition between prototypes using the synergetic potential derived in Chapter 2. Second, its structure agrees with theories on human pose estimation techniques in that it is based on transforming a novel image into a canonical view by performing a mental rotation.

As a practical pose estimation scheme, however, it has a number of serious shortcomings. Foremost amongst these are the facts that even relatively simple shapes such as the duck require a more sophisticated mental rotation procedure to be robust, and

that each image requires approximately 30 seconds to be processed.

It is worth noting that the concept of synergetic warping is not restricted to the particular transformation described here. By replacing this naive rotation transformation with a more sophisticated one, it should be possible to construct a more robust pose estimation routine. Alternatively, by using the rubber sheet transformation proposed by Daffertshofer and Haken [16], synergetic warping can be used to robustly identify hand-written characters.

Chapter 8

Synergetic Interpolation

8.1 Motivation

Synergetic warping for pose estimation is attractive as a concept because of the parallels between it and certain theories concerning the human visual system. As described in Chapter 7, however, as a practical scheme it has a number of disadvantages.

In this chapter we describe our second method of synergetic pose estimation which we call, *synergetic interpolation*, and which is designed to be a practical view-based pose estimation system.

Pose estimation by synergetic interpolation is based on the assumption that the order parameters capture enough pose information to allow us to estimate the pose of an object from these features alone. Now instead of interpolating between known images, we interpolate between known feature sets.

Synergetic interpolation is based on Murase and Nayar's successful view-based pose estimation routine [73]. They used principal component analysis on a set of training images to create an *object manifold* in an *object eigenspace*, parameterised by the pose variables. By interpolating on the manifold, they showed that it was possible to effectively estimate the pose of the same object from a single test image.

8.2 Concept

There are two spaces in the pose estimation problem. The pose space, PS, is a three dimensional space representing the three rotation parameters. The image space, IS, has a dimension equal to the number of pixels in the image, which is typically more than 100^2 dimensions. The goal of pose estimation is to find a mapping from IS to PS.

Finding such a mapping is made difficult because of the dimension of IS, so the first step in any technique must be the definition of a third, intermediate, feature space, FS.

Defining an intermediate space splits the initial problem into two subproblems. First, how to select the feature extraction technique and second, how to calculate the mapping from FS to PS.

8.2.1 Feature Space Design

As both PS and IS are givens, the design of FS is crucial to the success of the pose estimation routine.

Image space is capable of representing every possible monochrome image, so we denote the domain of IS in which images of the object of interest are found, as $\hat{IS}$. We label $\hat{FS}$ similarly as the domain of FS in which images of the object of interest are found.

The ideal FS would need to satisfy a number of criteria:

C1 FS should be of dimension such that

$$\dim(\mathrm{PS}) \leq \dim(\mathrm{FS}) \ll \dim(\mathrm{IS}). \tag{8.1}$$

C2 The size of $\hat{FS}$ should be maximised.

C3 There should be bijective mappings from $\hat{IS}$ to $\hat{FS}$ and from $\hat{FS}$ to PS.

The first criterion enforces the requirement that the feature space reduce the dimensionality of the problem. This goal is the main reason behind the introduction of FS. Typically, the dimension of FS is of the order of 10. The second criterion can be thought of as a weak form of the third criterion, because if it is met, the chances of meeting the third criterion are improved. If the third criterion is met, the introduction of feature space will not give rise to any ambiguity.

In this chapter we use the synergetic feature extractor from synergetic pattern recognition, as described in Chapter 2. Our features are the order parameters defined by Equations (2.18) and (2.21). As discussed in Section 6.1, Murase et al. [73], implemented this concept system using the scalar product of an image with the eigen-images as their feature extractor.

8.2.2 Explicit vs Implicit Mapping from Feature Space to Pose Space

Symbolically, the mapping from feature space to pose space can be written as,

$$\boldsymbol{p} = \boldsymbol{h}(\boldsymbol{\xi}) \tag{8.2}$$

where $\boldsymbol{p}$ is a 3-dimensional pose vector and $\boldsymbol{\xi}$ is an n-dimensional feature vector. This is the *explicit* model.

Now using our set of training images with known pose, we can calculate the corresponding $\boldsymbol{\xi}$ and thereby sample the function $\boldsymbol{h}$. Unfortunately, the nature of $\boldsymbol{\xi}$ makes it very difficult to interpolate between these known samples so as to estimate the pose of novel images. First, $\boldsymbol{\xi}$, has a variable, and possibly large dimension. This means that we have to interpolate on a high dimensional surface, which is computationally expensive. Second, the known values of $\boldsymbol{\xi}$ are not uniformly positioned over feature space. This adds a lot of algorithmic complexity to the task. Third, the size of the training set required to populate the space sufficiently for interpolation increases exponentially with the dimension of order parameter space. So if a 1-dimensional function requires a

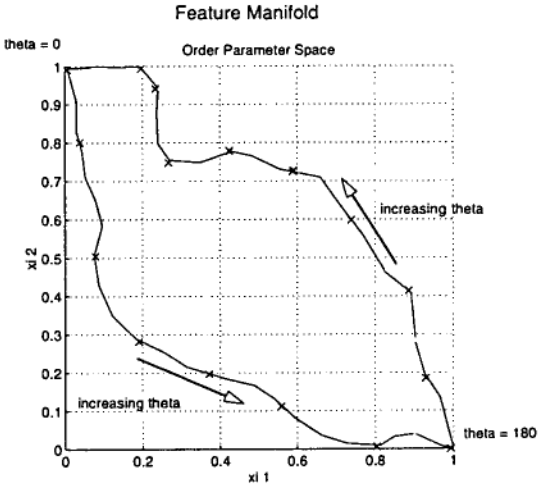


Figure 8.1: Illustrative feature manifold in two-dimensional order parameter space.

training images, an n-dimensional function will need of the order of a^n training images. With a typical dimension of 10, this is untenable.

To avoid these problems, we use an *implicit* model for pose estimation and approximate the inverse function

$$\boldsymbol{\xi} = \boldsymbol{h}^{-1}(\boldsymbol{p}). \tag{8.3}$$

Given that $\boldsymbol{p}$ is 3-dimensional, and that we can now choose the training set to be uniformly spaced over PS, the problems encountered with interpolating the explicit model are avoided.

To proceed we take our training images, which we now insist are sampled from a regular grid over the PS domain, and calculate the feature vectors for each. The result of this is an n-dimensional hyper-surface, parameterised by and sampled evenly over, the three independent pose variables. Every sample point on the hyper-surface represents a training pose. Now by fitting a surface through the sample points we create a hyper-surface which we call the *feature manifold*, or FM. FM represents the object at every possible pose and every point on FM represents a distinct pose.

The concept of a feature manifold is illustrated in Figure 8.1 which shows a manifold parameterised by a single pose variable in a two-dimensional order parameter space. The manifold is a smooth interpolation through the training points, each of which has an associated pose value.

Using the implicit model does, however, introduce two issues. First, our model ignores the interdependence of the $\boldsymbol{\xi}$ values, treating them as if they were truly independent. Second, our goal is to approximate $\boldsymbol{h}$, not $\boldsymbol{h}^{-1}$, so we must introduce another step which attempts to invert the function numerically.

Imagine that the training set contained one image of the object for every possible pose. Then no interpolation will be required and the feature manifold will be equal to the actual pose manifold in feature space. In this case, we could proceed by taking a test image and projecting it onto a test point in feature space. Because the feature

manifold is defined perfectly, the test point will lie on it, so we simply need to find the stored pose value associated with the matching point on FM, and we have found the required pose value.

In reality, the feature manifold will be an approximation to the actual pose manifold, because it is created by interpolating through a number of known points. We proceed in a similar vein by projecting the test image onto a test point in feature space. Now referring again to the feature manifold in Figure 8.1, we would expect the test point to lie near FM, since the manifold is an approximation to the actual pose manifold. We estimate the pose of the test image by finding the point on FM closest to our test point and interpolating along FM to find an estimated value for the pose at that point. This is the pose value we assign as our estimate for the test image. If the image is of a different object, we would not expect the test point to lie near the feature manifold. If this is the case, we can question the validity of the input image.

Referring again to the feature manifold in Figure 8.1, we estimate the pose of a test image by projecting the image into order parameter space. Assuming that it lies near FM, we find the point on FM closest to our test image and assign the associated pose value as our estimate.

8.3 Balance between Accuracy and Speed

Assuming that there is no innate ambiguity in the mappings from IS to PS, there are two possible causes for inaccurate pose estimation. The first occurs when FM is an inaccurate approximation to the actual pose manifold. The second is failing to find the global minimum of the distance from the novel point to FM.

It is clear then, that if we choose a large enough training set and find the global minimum of $d(\boldsymbol{p}, FM)$, the system will meet any possible error criteria. In practice however, neither of these assumptions are practical and the compromises that must be made on these issues will decide the balance between accuracy and speed.

A highly accurate surface definition is bulky and slow to search, but an inaccurate mapping will introduce errors. We must therefore choose a compromise position for the definition of our fitted surface. We have control over the number of data points used by the surface fitter. We also control the position of those points, with the restriction that they all lie on a regular grid of the independent variables.

Finding the minimum of $d(\boldsymbol{p}, FM)$ is also a cause of compromise as a search for the global minimum will be slow, but finding a local minimum instead will introduce possibly large errors. This problem is a case of non-linear multi-variable global minimisation and we have a number of options.

The first option is a tree-based search routine. The basic concept of this method is to store the data in specialised data structures based on an ordering of the data. These structures are often supplemented with indices that speed up the search process but add to the overhead costs.

There are many possible variations developed in the computer-science literature for fast database searching. The most appropriate choice for this problem is the search technique espoused by Nene et al. [76]. This is not guaranteed to find a global optimum. In fact it is not guaranteed to find a local minimum. It is, however, certain

to find a point close to (in terms of d) the global minimum and it does have excellent speed characteristics.

The second option is to use a gradient descent search directly on the surface defined by d. This is the classical solution to non-linear minimisation and there is a wealth of literature available. The search is for a local, not the global minimum, and the result of the minimisation is highly dependent on the starting point supplied by the user.

The correct choice depends on the particular application. In this chapter we test the algorithms on a number of different objects rotated around one axis. In this situation, the gradient descent method is inappropriate, because we have no basis on which to guess a starting point for the minimisation, and are likely to find large errors due to local minima. This, combined with the fact that we only need to search a single dimensional pose space suggests that we use a search-based technique.

In Chapter 10 we attempt to *track* the pose of a single object in time. Now, any physical object can only rotate a limited amount within a small timeframe so knowing the pose at a certain time gives a good starting position for the minimisation in the next time step. This fact can be used by the gradient descent method, but not by the tree-based search. As well as giving increased accuracy, the contiguity of the physical system will increase the speed of estimation because the estimation time decreases when the initial pose estimate is close to the actual pose.

A further advantage of the gradient descent based option is storage. A b-spline representation is capable of representing a surface in a much more compact fashion than an ordered database of surface points, as required by the tree-based search algorithm. Thus, for the same data storage usage, a b-spline representation is capable of recording a significantly larger section of the hypersurface.

8.3.1 Gradient Descent Algorithm

At the core of the gradient descent algorithm is the cubic b-spline surface fitting procedure. We need to fit a surface through a series of points with three independent variables and an arbitrary number of dependent variables. This is no easy task, and requires a very sophisticated surface fitting procedure. We used the DT_NURBS [5] library, available from the United States Navy, to implement this fitting. Note that this process is only carried out once during training, so a lengthy fitting procedure is quite satisfactory.

Having constructed and stored the feature manifold, the second key function must find the point on the surface closest to any given point in feature space. This function is called every time an estimate is required, so an efficient and accurate system is important. We have extended the DT_NURBS library with a fortran function which can achieve this with an arbitrary number of dependent and independent variables and works directly on the data structure used by the DT_NURBS library, making it as fast as possible. It is based on Gauss' method, and has been supplemented by the modifications suggested by Marquardt [66].

Given a point in feature space, $\hat{\xi}$ and a feature manifold, MF, our goal is to find the pose $\hat{p}$ such that,

$$||MF(\hat{p}) - \hat{\xi}|| = \min_{p}||\mathrm{MF}(p) - \hat{\xi}||, \tag{8.4}$$

where $||.||$ represents the Euclidean norm. Starting with the initial guess $\boldsymbol{p}_0$, the program calculates a sequence of (hopefully) improved estimates until it either converges or reaches a maximum number of iterations. The values defining the convergence and maximum iteration criteria are given by the user and are described below.

At each time step, A is the matrix of first partials of the feature manifold evaluated at $\boldsymbol{p}_i$, and $\boldsymbol{r}$, the vector of residuals, is given by $\boldsymbol{r} = \hat{\boldsymbol{\xi}} - MF(\boldsymbol{p}_i)$. The next iterate is calculated as $\boldsymbol{p}_{i+1} = \boldsymbol{p}_i + h \times \boldsymbol{d}_i$, where $\boldsymbol{d}_i$ is the least squares solution to $A\boldsymbol{d}_i = \boldsymbol{r}$.

When the step-size, h, is arbitrarily defined, we have Gauss' method. Marquardt realised that there is no guarantee that a value of h exists which will ensure that the next iterate is an improvement over the current one, so he suggested the following modifications.

The matrix equation $A^T A\boldsymbol{d}_i = A^T\boldsymbol{r}$ which is used to find $\boldsymbol{d}_i$ is replaced with,

$$A^T A\boldsymbol{d}_i + \gamma I = A^T\boldsymbol{r}, \tag{8.5}$$

where γ is a user supplied positive constant and I is the identity matrix. Now when γ is large, $\boldsymbol{d}_i$ rotates towards the direction of steepest descent, so for a large value of γ and a small value of h, improvement should occur. Clearly however, we would like the step size to be as large as possible, while still improving the estimate, because a very small h will increase time costs.

We implemented a heuristic approach [5] to balancing the values of γ and h. Introducing some more user defined constants, the heuristic is,

1 $\gamma = \gamma_0$ where subscript 0 designates the default value;

2 $h = h_0$ where subscript 0 designates the default value;

3 if current residual is less than residual for previous estimate, stop.

4 if number of halvings equals max_half, goto **6**;

5 $h = h/2$, goto **3**;

6 if number of gamma multiplications equals max_amp, stop.

7 $\gamma = \gamma \times$ amplification factor,

8 goto **2**.

We also need parameters to define the stopping criteria. The user must set a maximum number of iterations, after which the system is considered to have failed to converge on an answer. The first success criterion is when the residual is sufficiently small,

$$||\boldsymbol{r}_i|| < \epsilon_1. \tag{8.6}$$

The second is when the relative change in the residual is small,

$$\frac{||\boldsymbol{r}_{i+1} - \boldsymbol{r}_i||}{||\boldsymbol{r}_i||} < \epsilon_2. \tag{8.7}$$

The third stops the iterations when the absolute change in the pose estimate is small,

$$||\boldsymbol{p}_{i+1} - \boldsymbol{p}_i|| < \epsilon_3, \tag{8.8}$$

and the fourth when the relative change falls below a user defined limit,

$$\frac{||\boldsymbol{p}_{i+1} - \boldsymbol{p}_i||}{||\boldsymbol{p}_i||} < \epsilon_4. \tag{8.9}$$

All of the user defined parameters, along with minimum, maximum and recommended values where appropriate, are listed in Table 8.1.

Variable	Use	Min	Max	Recommend
γ	Marquardt factor	0	1	.25
a	amplification factor	1	10	2
max_half	max number of h halvings	2	8	3
max_amp	max number of Marquardt amplifications	4	20	5
max_iter	max number of iterations	1	na	30
ϵ_1	min residual	problem dependent		
ϵ_2	min relative change in residual	problem dependent		
ϵ_3	min absolute change in pose estimate	problem dependent		
ϵ_4	min relative change in pose estimate	problem dependent		

Table 8.1: Values for Gauss/Marquardt minimisation routine.

8.4 Examples

To compare synergetic interpolation to synergetic warping, we now test the system on the objects shown in Figure 7.4. For consistency with the results reported in Table 7.1, we again have 12 training images rotated in 30° steps from one another. As described above, we must decide on the optimum number of these to select as prototypes. To reflect the different possibilities, Table 8.2 records the results across a range of different prototype numbers.

The results given in this table are illustrative of a number of issues discussed above.

We concentrate at first on the results for the duck. Note the strong effect that the number of prototypes has on the error, with the optimum choice being four prototypes. For fewer prototypes, our original assumption that the order parameters will capture enough of the pose information to be able to estimate pose, is incorrect. This is illustrated in Figure 8.2(a), which shows the two dimensional order parameter space for the duck. As the interpolation is based upon this curve, not on the original images themselves, the fact that the curve is self-intersecting shows that our projection has introduced artefact ambiguity. This occurs when two or more dissimilar images are projected onto nearby locations in order parameter space, and results in

used by Murase et al [73]. In this case the system used all of the training images to calculate the required number of eigen-images.

We applied all three options to the twenty objects in the COIL database [77] as the basis for a numerical comparison. Each object was described by a training set of 9 images rotated in 30° steps, each of which was used to define the manifold in order-parameter space. For each option, we experimented with a range of prototypes, only the best of which is reported in Table 8.3.

Object	Synergetic	Orthonormalised Synergetic	Eigen-Image
Duck	1.28	1.73	1.08
Wooden Block 1	2.17	1.75	1.02
Car 1	4.37	2.41	1.48
Cat	2.69	2.18	1.27
Toothpaste	7.10	4.68	1.72
Car 2	15.3	19.8	7.41
Wooden Block 2	2.18	1.42	1.49
Talcum Powder	2.46	1.31	1.15
Tylenol	26.8	22.0	19.6
Vaseline	3.29	2.24	1.02
Wooden Block 3	5.96	4.86	1.37
Japanese Cup	6.66	7.44	8.17
Piggy Bank	4.95	3.76	2.25
Gasket	7.22	8.13	4.73
Salad Spinner	4.64	3.08	3.55
Hair Conditioner	7.44	6.85	7.02
Bowl	8.88	8.24	10.4
Tea Cup	9.53	5.36	10.1
Car 3	33.2	24.3	25.3
Cream Cheese	7.53	7.28	5.24

Table 8.3: Mean estimation errors in degrees from interpolating over various feature extractors.

The most notable factor in this table is the variance between objects. Inspection of those objects with large mean error values shows that they suffer from *view ambiguity*, leading to *feature ambiguity*. With the exception of the toothpaste, a different choice of feature extractor was unable to resolve the ambiguity.

Nonetheless, it is clear that the choice of feature extractor can alter the accuracy of the resulting estimation. This effect is not uniform across all objects but, in general, the best results were found using the eigen-image, followed by the orthonormalised prototypes.

8.5.1 Pose Space Subdivision

One approach to the problem of view ambiguity is subdividing pose space into a number of smaller domains, within which there is no ambiguity. For the Tylenol package with three order parameters as shown in Figure 8.2(b), we can subdivide pose space into the domains $0 - 179°$ and $180 - 359°$.

If we can successfully decide to which of these two subdomains any given image belongs, then interpolating within the appropriate subdomain yields over the entire domain, an average error of 5.67° and a maximum of 13°. This level of error is in stark contrast to the results quoted in Table 8.2.

The question remains as to how we can determine to which subdomain a given image belongs. A successful determination will lead to significant reduction in error values due to view ambiguity. However, an incorrect determination will introduce its own errors because an image classified to the wrong subdomain will never be classified correctly, even if the global minimum is correctly situated for that example. There are three options available to us when trying to distinguish between ambiguous views.

The first is to use the entire image information, not simply the order parameter values. Since the pose of the Tylenol package was successfully estimated using synergetic warping, it is clear that these ambiguities can be clarified, and each image successfully classified using the appropriate subdomain, by using the full image information. Unfortunately, this process significantly increases the memory requirements of the system, which must now also store the entire training set of images.

The second option is to define a new set of order parameter values that are designed specifically to distinguish between the ambiguous views. Fortunately, the means to do this are already well established in the form of the MELT algorithm. We simply take the training set of images, classify them depending upon the appropriate subdomain, and use MELT to create the appropriate adjoint prototypes. While simple and inexpensive on memory, this approach will often fail to find a way of completely separating the two domains because of the innate view ambiguity which any projection will find difficulty in overcoming. In the particular instance of the Tylenol package, 14 of the 72 images were mis-classified. Most of these errors were near the subdomain boundaries, and therefore lead to relatively small pose estimation errors.

The final option is to use object-specific information to clarify the situation. If a human was asked to distinguish between the two images shown in Figure 8.3, the response would probably involve the orientation of the brand name. This is object-specific information which we use to clarify ambiguous situations. One possible solution then, is to use optical character recognition to distinguish between these two options. The smaller sides of the box could be distinguished by the existence, or otherwise of a barcode.

8.6 Conclusions

Synergetic interpolation is a fast, practical object-independent approach to pose estimation. It is based on the assumption that the features extracted from the image capture enough pose information to interpolate pose on the resulting manifold, without introducing any feature ambiguity.

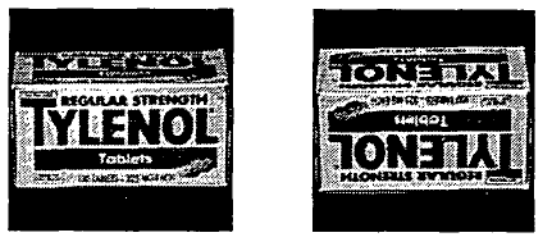


Figure 8.3: Two views of an object showing view ambiguity.

The results reported here show a large range of error values, dependent on the shape of the object. Those objects with view ambiguity are poorly estimated, and require the use of object specific information or pose-space subdivision to resolve the ambiguity.

It is clear that the major design decision in this method is the choice of feature extractor, which leads us to the next chapter in which we ask if we can find an optimal feature extractor.

Chapter 9

Explicit Inversion

9.1 Motivation

We have strived in this work to look for the most general approach to pose estimation. We therefore rejected the use of CAD models in favour of view-based pose estimation. We have extracted generic features from images, without any specific knowledge of the object involved. Even beyond this, synergetic interpolation as described in Chapter 8 can be used to estimate any continuously valued function, such as the direction of lighting in the image.

In this chapter, we take our broadest view yet. Here we introduce a new approach to *signal analysis*, of which both pose estimation and pattern recognition are special cases. Given a signal, which is the output of some complicated and possibly unknown function, the goal of signal analysis is to estimate the parameters of that function. Formally, at least, the solution to the problem can be found by inverting the function which produced the signal. In practice this inversion requires two major elements; a feature extractor to limit the dimension of the signal and a parameter estimator. The reader will recall that these are the same two elements required by the synergetic interpolation method of Chapter 8. While there has been much research into these two elements, they are generally designed separately from one another, whereas recognition of the relationship between these two elements suggests that they should be designed as a pair. Following this design concept allows us to replace the problem of implicitly inverting an unknown, possibly high-dimensional function, with that of explicitly inverting a known, low-dimensional function. Amongst other benefits, the major advantage of following this method is a dramatic increase in speed over the standard approaches.

To be more concrete, we will restrict our discussion to the challenge of *image analysis*, to which our approach is particularly well suited due to the size of the signal vector. It should be kept in mind, however, that our technique can be applied to any form of signal.

9.2 Problem

Consider a set of n grey-scale images, each of which has been sampled so as to have l pixels. We store each image $\boldsymbol{q}_i$ as a column vector in a $l \times n$ matrix, labelled Q. Throughout this chapter we will assume that all image vectors have been scaled to have unit length.

These images are the training set for our image analysis. If our task is qualitative, each image in the training set is associated with a class $c_i \in \mathbb{N}$, to which the image belongs. If our task is quantitative, each training image has a parameter vector $\boldsymbol{p}_i \in \mathbb{R}^u$ associated with it. In general, these parameters will not represent quantities that are directly measurable from the image. Rather each element of $\boldsymbol{p}_i$ will be a physical characteristic of the object or objects in the image where the relationship between the characteristic and the pixel values is complicated and unknown.

Given a test image $\boldsymbol{q}$, our goal is to return either a classification c or a parameter vector $\boldsymbol{p}$, which correctly matches or closely approximates, the correct values for the test image.

Clearly a pixel-by-pixel comparison between Q and $\boldsymbol{q}$ is unwieldy and expensive. For this reason we introduce a feature extractor $\boldsymbol{f}$ to calculate m features $\boldsymbol{\xi} \in \mathbb{R}^m$,

$$\boldsymbol{\xi} = \boldsymbol{f}(\boldsymbol{q}). \tag{9.1}$$

Applying this to the n training images, we store the resulting n feature vectors, $\boldsymbol{\xi}$, in an $m \times n$ matrix, Ξ. When presented with a test image, $\boldsymbol{q}$ we apply the same feature extractor to calculate $\hat{\boldsymbol{\xi}}$, which is the comparable set of features for the test image.

Next we need to construct a classifier or parameter estimator, $\boldsymbol{h}$, which estimates the class or parameter values from the extracted feature vector,

$$\boldsymbol{p} = \boldsymbol{h}(\boldsymbol{\xi}) \text{ or } \mathrm{c} = \boldsymbol{h}(\boldsymbol{\xi}). \tag{9.2}$$

The complete flow of an image analysis system is shown in Figure 9.1 and Equation 9.3,

$$\text{parameter/class} \xrightarrow{\boldsymbol{g}} \text{image} \xrightarrow{\boldsymbol{f}} \text{features} \xrightarrow{\boldsymbol{h}} \text{estimated parameter/class}. \tag{9.3}$$

Now since the imaging function $\boldsymbol{g}$ is fixed, both of these flow diagrams show clearly that we should choose $\boldsymbol{f}$ and $\boldsymbol{h}$ to be related. Indeed we can state that the optimal choice of functions would give a perfect image analysis system where,

$$\boldsymbol{p} = \boldsymbol{h}(\boldsymbol{f}(\boldsymbol{g}(\boldsymbol{p}))) \text{ or } \mathrm{c} = \boldsymbol{h}(\boldsymbol{f}(\boldsymbol{g}(\mathrm{c}))). \tag{9.4}$$

Yet the standard approach is to select a feature extractor and an estimator/classifier separately. We will see that by designing functions $\boldsymbol{f}$ and $\boldsymbol{h}$ as a pair, we can produce major time and memory savings.

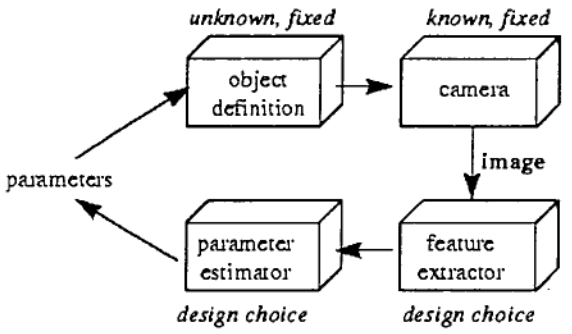


Figure 9.1: Schematic of a parameter estimation problem.

The rest of this chapter is structured as follows. First we look at feature extractors. We introduce a set of ideal characteristics for feature extractors, and use them to compare a number of popular feature extractors in Section 9.3. In Sections 9.4 and 9.5 we review a number of standard image classifiers and parameter estimators respectively. Section 9.6 introduces our method for constructing extractor/classifier and extractor/estimator pairs and describes how to design these pairs to solve specific problems. In Section 9.7 we look at a number of examples which clearly show the advantages of our new approach over the standard method before drawing our conclusions in Section 9.8.

9.3 Feature Extractors

Feature extractors can be categorised as object-specific or object-independent. An example of an object-specific feature extractor is an algorithm that measures certain distances in images of a human face. While features such as eye-to-eye distance are used successfully to implement face recognition they do have a number of drawbacks. First, locating individual parts of the face, such as the eyes is a challenging task [68]. Second, as the features are object specific, effort invested in algorithms to extract them cannot be easily transferred to other objects.

All of the feature extractors described below are object-independent. They treat the image as a whole and do not attempt to extract information which is specific to particular objects.

Feature extractors may also be classified as either problem-specific or problem-independent. As the title suggests, a problem-specific feature extractor is designed to achieve a particular image analysis task. This can be represented symbolically by including the associated class or parameter values in the definition of the feature extractor,

$$\xi = f(q, c) \quad \text{or} \quad \xi = f(q, p). \tag{9.5}$$

Problem-independent feature extractors on the other hand work solely on the images in the training set without reference to p or c.

All of the feature extractors described here share a similar construction. We extract m features by calculating the scalar product of an image with m images known as *prototypes*. Each prototype is a linear superposition of the training images, Q, such that,

$$\xi = f(q) = Vq = GQ^T q, \tag{9.6}$$

where V are the prototypes, G is an $m \times n$ linear superposition matrix and the superscript T denotes the matrix transpose operation. The difference between the various feature extractors described below, is in the construction of G.

9.3.1 Comparing Methods

When deciding between various possible G matrices, a useful measure is the ability of the system to reconstruct the training set images. The better the reconstruction, the more 'information' has been extracted from the training set.

If G is the identity matrix, there is no reconstruction error, but there is also no data reduction. So in practice we look for a G which reaches a useful compromise between data reduction and reconstruction error.

An optimal feature extractor of the type shown in Equation 9.6 would have the following characteristics:

C1 a robust, deterministic procedure to calculate G; and

C2 a low reconstruction error with reasonable data reduction.

Depending on the type of image analysis task, it would also project the images into feature space such that:

C3a the classes could be easily separated; or

C3b the features populate a smooth manifold upon which interpolating the parameter vector p is simple.

9.3.2 Review of Feature Extractors

Correlation

The simplest approach to the construction of G is for the user to select a subset of the training images as the prototypes. The associated matrix G_c is zero everywhere except for a single 1 in each row. No column has more than a single non-zero entry. As only the image data is used, the correlation method is a problem-independent approach.

The resulting feature vector yields the correlation between the test image and the prototypes. This is a measure of the distance between the test images and the prototypes in the high-dimensional image space.

In the case that all of the training images are used, G_c simply becomes the identity matrix of size n and the approach fails to meet **C2**. We must store the entire set of training images and calculate the correlation with each. In the literature, implementations of this approach have required the development of special purpose VLSI hardware to deal with the size of this calculation [29].

For most practical problems $m < n$, so the user must choose a subset of the training set which is representative of the entire set. This task is known as *subset selection*. A number of alternatives for making an appropriate decision exist, including QR decomposition [32], SVD decomposition [31] and total least squares fitting [53].

Of the three optimal characteristics, this approach satisfies **C1** through the use of the subset selection algorithm. **C2** may also be met, but the more sophisticated feature extractor designs shown below are capable of producing lower reconstruction errors with the same level of data reduction. The question of whether **C3** is met, can only be answered on an individual case basis.

Synergetic Feature Extractor

The features calculated by the synergetic feature extractor are the order parameters defined by Equations (2.18) and (2.21). The synergetic feature extractor is problem-specific, and is derived from multivariate linear regression [55], so as to minimise the reconstruction error [79].

We have already seen the synergetic feature extractor used for image classification in Chapters 2, 3 and 4, and for parameter estimation in Chapter 8.

We now re-formulate the synergetic feature extractor into the formulation of Equation (9.6). As with the correlation method described above, we select one image from each class to be the class prototype by defining the matrix G_c. Now recall that the synergetic feature extractor is defined such that the kth prototype is projected onto the kth unit axis in order parameter space. This means that the matrix of training features, Ξ, must be the identity matrix I, so we need to define G_s such that,

$$\Xi = G_s Q^T V = I, \tag{9.7}$$

where $V = QG_c^T$ is the prototype matrix.

This requirement is met when G_s is given by

$$G_s = (G_c Q^T Q G_c^T)^{-1} G_c, \tag{9.8}$$

as can be confirmed by substituting Equation 9.8 into Equation 9.7,

$$\Xi = (G_c Q^T Q G_c^T) G_c Q^T Q G_c^T = I. \tag{9.9}$$

This new matrix formulation of the synergetic feature extractor highlights the previously unreported fact that the synergetic feature extractor is equivalent to Kohonen's Optimal Linear Identification [60].

The scheme can be extended to allow multiple training images per class [12] using the MELT algorithm described in Chapter 6. In this case, all training images assigned to the ith class are projected onto unity on the ith axis in feature space.

To fulfil **C1**, we must complete the matrix inversion in Equation 9.8. This requires that the training set be linearly independent, which is not in general a problem for images, but can be restricting for shorter input vectors.

As G_s is a function of G_c, the reconstruction error will again be dependent on the prototypes chosen by the subset selection method. However, as discussed in Section 2.4, it has been shown [79] that given a particular G_c, G_s provides the minimum possible reconstruction error. Thus the synergetic feature extractor compares favourably with the correlation method when judged on the basis of **C2**.

Judged solely on the training set, the multiple training image per class synergetic feature extractor fulfils the requirements of **C3a** by definition. The m-dimensional feature space contains just m points, one at unity on each axis, and so separating the classes is trivial. If the training set is representative of the likely test data, this separability is likely to be maintained.

The single training image construction is more likely to meet **C3b**, because extracting the same features from two different images creates a multi-valued function, making interpolation difficult. Intuitively, the fact that the reconstruction error is minimal for any given G_c suggests that the synergetic feature extractor is more likely to meet **C3b** than the correlation method.

Eigen-Images

Eigen-images are a popular form of prototype for both qualitative [94, 98, 99, 42] and quantitative [73, 74] image analysis. Their popularity is due to the following useful characteristics. First, the prototypes are designed to capture as much of the variation among the training set as possible. This means that significantly fewer prototypes are needed to represent the training set, which reduces memory and time costs in comparison to the previous methods. Second, the eigen-images are orthogonal, making it simple to reconstruct the original image from the eigen-images.

The eigen-images are in fact the eigenvectors of the image covariance matrix, which are the result of applying Principal Component Analysis (PCA) [55] to the training images Q. In our formalism, G_e is a matrix with orthonormal rows that rotates the axes so as to maximise the variance over the axes. Now a measure of the variance on each axis is supplied by the corresponding eigenvalue, and so we can choose the m eigenvectors with the largest eigenvalues as the prototypes. In this way we maximise the amount of 'information' captured in the prototypes. Clearly if we choose all n eigenvectors we will be able to reconstruct each training image perfectly. In terms of memory and time requirements, however, we will have reverted to the simple correlation procedure.

Eigen-images satisfy **C1** because of the availability of PCA routines. For large training sets, considerable time savings can be obtained by using algorithms that only calculate the eigenvectors associated with the largest m eigenvalues [72]. For online learning, update PCA algorithms are also available [65].

Eigen-images have excellent properties when measured against **C2**. The previous methods required one prototype for each image in the training set, and we were therefore forced to select a subset of the training images, thereby ignoring possibly valuable information. The eigenvector approach allows the user to present the entire training set and then select how many prototypes are required to reach the desired reconstruction error. In this way, the user can control the compromise implicit in **C2**.

Unfortunately, problems arise with **C3** because the approach is problem-independent. By maximising the variance over the entire set, it is likely that the features will populate a smooth manifold in feature space, thereby partially fulfilling **C3b**. However, as the prototypes were not designed using the associated parameters $\boldsymbol{p}_i$, we cannot judge the ease of interpolating the parameter vector $\boldsymbol{p}$ on the resulting manifold. Neither are the eigen-images likely to fulfil **C3a** as the smooth manifold will have, in general,

smeared different classes together, making class separation difficult.

Fisher-Images

Fisher-images are a generalisation of eigen-images, which provide a problem-dependent feature extractor, designed specifically for image classification. The eigen-image based feature extractor attempts to maximise the variance of the features, in a problem-independent fashion. For an image classification task, the training class information is ignored, and it is quite possible that classes which were previously linearly separable will be projected into the same section of subspace. Clearly, this is not ideal for image classification.

Fisher-images, on the other hand, are designed to maximise the inter-class variance while minimising the intra-class variance. A construction which achieved this would tend to have small, isolated clusters of same-class points in feature space, leading to a simple decision boundary.

This is the idea behind the construction of Fisher-images [7], which are a variation on Fisher's Linear Discriminant [25], and are calculated using a generalisation of PCA. Again, G_f is a matrix with orthonormal rows, where for Fisher-images, it maximises the ratio of inter-class variance to intra-class variance.

Fisher-images satisfy **C1** and **C2** for the same reasons as given for eigen-images. The reconstruction error will no longer be minimal because of the requirement to have low intra-class variance but the level of compromise is still in control of the user who selects the number of prototypes. They do, however have an important restriction that for a d class problem, the maximum number of prototypes is $d-1$ [21]. Unlike eigen-images, this is an upper limit on how many prototypes can be used to minimise the reconstruction error.

As stated above, Fisher-images are designed specifically to meet **C3a**. They are unsuitable for **C3b** because the concept of a class does not exist in quantitative tasks.

9.4 Image Classification

The task of image classification is equivalent to approximating the function h in Equation 9.4. Having extracted features, Ξ and a test point $\hat{\xi}$, we must now use a classification scheme to associate the correct class with the test input image. If the feature extractor has been well designed, a test image of class i will be projected into feature space near points from the training set of class i. Thus a distance measure in the feature space forms the basis on which we will make our classifications. For a standard guide to image classifiers, the reader is referred to Duda and Hart [21]. We will look briefly at three popular classifiers: nearest neighbour, k-nearest neighbour and linear-discriminant functions.

9.4.1 Review of Classifiers

Nearest Neighbour Classifiers

In the nearest neighbour scheme, we calculate the distance in feature space from the test point to each training point. If the closest training point has classification c_i, then

we classify the test image as belonging to class i.

The nearest neighbour classification scheme is an example of a non-parametric classifier, in that we do not pre-suppose a parameterised shape for the class boundaries. This leads to a very flexible boundary which is capable, for example, of successfully dealing with a class which is broken into distinct, separated clusters. However, it can yield poor results when combined with a problem-independent feature extractor such as correlation and eigen-images because no attempt has been made to cluster the data.

While simple in concept, it can also be an expensive scheme to compute. In a naive implementation, we must calculate the distance to every training point in order to find the closest. It is possible to speed up the search by implementing a sophisticated search structure, such as a tree or indexed list structure [76].

The k-nearest neighbour scheme is an extension of the nearest neighbour scheme in which the k closest points are found. The test image is classified as belonging to the median class in the k closest points.

Like its simpler cousin, the k-nearest neighbour is capable of creating flexible decision boundaries. It also has the advantage of being less likely to deteriorate in the presence of poorly clustered data. However, we must find the k closest points, which exacerbates the computational problems confronted by the simple nearest neighbour routine and the choice of k is an extra parameter which must be set by the user.

Linear Discriminant Functions

Linear discriminant functions are a parametric classifier with the form,

$$h_i(\hat{\boldsymbol{\xi}}) = \boldsymbol{w}_i^T \hat{\boldsymbol{\xi}} + w_{i0}, \qquad \forall i = 1, \ldots, s, \tag{9.10}$$

where s is the number of classes.

The values for $\boldsymbol{w}_i$ and w_{i0} are calculated based on the training set and test images are classified using the following rule,

$$\begin{array}{ll} h_i(\hat{\boldsymbol{\xi}}) > h_j(\hat{\boldsymbol{\xi}}) \;\; \forall j \neq i & \Rightarrow \;\; c = c_i \\ \text{if no such } i & \Rightarrow \;\; c \text{ undefined} \end{array} \tag{9.11}$$

The resulting classifier splits feature space into s regions bounded by the hyperplanes, $h_i(\hat{\boldsymbol{\xi}}) = h_j(\hat{\boldsymbol{\xi}})$.

The linear discriminant function classifier is only capable of separating linearly separable clusters. The simplicity in the structure allows most of the computational work to be done during training when calculating the weights, $\boldsymbol{w}$. As a result, test points can be classified quickly.

9.5 Parameter Estimation

The task of parameter estimation is equivalent to approximating the function $\boldsymbol{h}$ in Equation 9.4. Having extracted features, Ξ and a test point $\hat{\boldsymbol{\xi}}$, we must now interpolate over the Ξ with known $\boldsymbol{p}_i$, to estimate the unknown $\boldsymbol{p}$ of the test image. As was the case with image classification, the distance between points in feature space is the appropriate measure for comparing images.

9.5.1 Review of Parameter Estimation Methods

Search

The search based methods for parameter estimation are equivalent to nearest neighbour classification. Again we look for the point in feature space that is closest to our model point. We then estimate $\boldsymbol{p}$ with the $\boldsymbol{p}_i$ value associated with that point. Recalling that $\boldsymbol{p} \in \mathbb{R}^u$ and recognising the fact that the system can only return values of $\boldsymbol{p}_i$, it is clear that we must have a large training set for $\boldsymbol{p}$ to be estimated accurately. This exacerbates the time costs previously described for nearest neighbour classifiers and makes it even more necessary to use the advanced search techniques referenced above. Thus for search-based methods, the user must decide on a compromise between accuracy and speed. A combination of eigen-images and an advanced search technique has been used for problems in pose and lighting direction estimation, and forms the basis of a commercially available system [73, 74].

Minimisation

The search based methods do not take advantage of the fact that a well designed feature extractor should project points onto a smooth manifold within feature space. Using this fact, we can design a gradient-descent based routine that will find local minima in the distance-to-surface function.

Just such a gradient-descent routine was described in detail and used to estimate pose in Chapter 8.

The first step in this process is to construct a hyper-surface through the points in Ξ using a flexible tool such as B-splines. Each point on the surface is defined by m features and is parameterised by the u elements of $\boldsymbol{p}$. Now that we can calculate both the distance and the gradient of the distance to the surface with respect to the parameter vector $\boldsymbol{p}$, we can iteratively improve an initial guess $\boldsymbol{p}_0$ until it reaches a local minimum.

The advantage of this technique is that the constructed surface completely parameterises the vector $\boldsymbol{p}$, thereby allowing the system to return values for $\boldsymbol{p}$ not present in the training set, but interpolated between them. Unfortunately, the need to construct such a surface places limits on the choice of data in the training set. In order to build a high-dimensional surface parameterised by a possibly large number of variables, most surface fitting routines require that the data be situated on an evenly spaced grid. One cannot therefore, give more training examples in an area of parameter space where problems are likely to arise without doing the same in areas which should be simple to interpolate.

The time costs for the parameter estimation are no longer linked to the size of the training set, but rather on the accuracy of the initial guess. So too is the accuracy of the estimate. A poor initial guess may lead the system into a false minimum, causing large approximation errors.

This technique is most useful when a good initial guess is available. One such application is that of tracking the pose of an object through a video sequence, where the availability of the result from the previous frame reduces time and error in the

estimation of the current frame [45, 47]. This problem will be discussed further in Chapter 10.

9.6 Explicit Inversion

We now reconsider the problem of image analysis in the light of our belief that the feature extractor and the classifier/estimator should be designed as a pair. As a result, we propose a new method which we call 'explicit inversion' [46], because it replaces the implicit inversion carried out by the current classifier/estimator algorithms with an explicit, algebraic inversion.

9.6.1 Designing a Feature Extractor

We start by recalling our definition of the perfect image analysis system, Equation 9.4.

When using the correlation, eigen-image or fisher-images based feature extractors, it is clear that most of the responsibility for achieving this goal is concentrated in the design of the estimator $\boldsymbol{h}$.

We argue that this responsibility should be moved to the feature extractor $\boldsymbol{f}$. In this way $\boldsymbol{h}$ can be greatly simplified, leading to a reduction in classification/estimation times.

This is achieved by designing a feature extractor which can return an arbitrarily defined set of feature vectors for the images in the training set.

From Equation 9.6 we can state that the training set features are given by

$$\Xi = G_p Q^T Q. \tag{9.12}$$

Now we wish to find G_p such that Ξ can be designed arbitrarily by the user. We can achieve this by re-arranging to make G_p the subject,

$$G_p = \Xi (Q^T Q)^{-1}. \tag{9.13}$$

For a given Ξ, the resulting feature extractor is Kohonen's Optimal Linear Associative Mappping [60], re-expressed in terms of our standard form for feature extractors. The key to successfully applying our approach is the design of Ξ and we refer to the resulting feature extractor as a *designed feature extractor*. We show how to design a feature extractor that satisfies **C3a** for image classification in Section 9.6.2. The requirement of **C3b** for parameter estimation can also be met by a good design of Ξ, as explained in Section 9.6.3.

In order to satisfy **C1**, the correlation matrix $Q^T Q$ must be non-singular. As discussed above, the sheer length of image vectors, combined with the normalisation assumptions in place mean that the training images are almost certain to be linearly independent. When this is not the case, or for application to shorter signals where linear independence is not so likely, the Moore-Penrose pseudo-inverse supplies a least-squares solution, as described in Appendix B.

In general, designed feature extractors will fail criterion **C2** because no attempt has been made to minimise the reconstruction error. However, the reasoning behind **C2** is that the likelihood of ambiguity within feature space on an unknown distribution

in feature space is lessened if the reconstruction error is small. With a designed feature extractor, however, we know the exact distribution of the training set in feature space, so **C2** is of secondary importance. Nonetheless, the reconstruction error does play a role in applying explicit inversion, as will be shown in the examples of Section 9.7.

It is worth noting that the synergetic feature extractor is a special case of a designed feature extractor. In the general case, the training set can be projected onto *known arbitrary* points in feature space, while the synergetic feature extractor projects the training set onto the unit axes.

9.6.2 Direct Image Classification

We now have the tools to design a feature extractor which extracts arbitrary features from the training set. In this section we see how we can use these to design a feature extractor/classifier pair for image classification.

When the distribution of points in feature space is unknown, we are forced to choose between a powerful but slow non-parametric classifier or a less-powerful and fast parametric classifier. This is because the function $\boldsymbol{h}$ is attempting to invert an unknown, complicated and high-dimensional function.

In contrast, our designed feature extractor allows us to create a classifier with the speed of linear discriminant functions and the effective classification power of the nearest neighbour classifiers.

To do this we design Ξ such that,

$$\boldsymbol{\xi}_i = \boldsymbol{f}(\Xi, \boldsymbol{q}_i, c_i) = \boldsymbol{h}^{-1}(c_i). \tag{9.14}$$

Now, as our choice of notation implies, by choosing $\boldsymbol{h}^{-1}$ to be a simple, analytically invertible function, we can trivially find the required classification function $\boldsymbol{h}$.

In a simple two class example, we might extract a single feature and create a projection such that,

$$\begin{aligned} c_i &= 1 \Rightarrow \xi = -1, \\ c_i &= 2 \Rightarrow \xi = 1. \end{aligned} \tag{9.15}$$

Now inverting this function we find our classifier $\boldsymbol{h}$,

$$\begin{aligned} \hat{\xi} &< 0 \Rightarrow c = 1 \\ \hat{\xi} &> 0 \Rightarrow c = 2 \\ \hat{\xi} &= 0 \Rightarrow c \text{ undefined}. \end{aligned} \tag{9.16}$$

Inspection of this example makes it clear that all three classifiers; nearest neighbour, k-nearest neighbour and linear discriminant functions, would calculate the same classification boundary. Thus our choice of Ξ has made all three classifiers equivalent.

As the synergetic feature extractor is a special case of a designed feature extractor, we can also classify a test image directly from $\hat{\boldsymbol{\xi}}$, albeit without the flexibility offered

by an unrestrained choice of Ξ. For the synergetic feature extractor, the classification rule is,

$$\begin{aligned} |\hat{\xi}_i| > |\hat{\xi}_j| \ \ \forall i \neq j \quad &\Rightarrow \quad \text{class} = \text{i} \\ \text{if no such } i \quad &\Rightarrow \quad \text{class undefined.} \end{aligned} \tag{9.17}$$

9.6.3 Direct Parameter Estimation

When the distribution of points in feature space is unknown, we are forced to choose between a fast system with small training requirements which only finds local minima or a slower system which finds the minimum of a large set of training examples but is incapable of interpolating between these examples.

Our designed feature extractor means that we do not need to accept either of these compromises. A good choice for the locations of the training points allows us to create a parameter estimation system which is significantly faster than either of the current methods, which requires only a relatively small training set but which can effectively utilise a large, unevenly distributed training set and which is guaranteed to find the global minimum of the distance-to-surface function.

To do this, we design Ξ to fit $\boldsymbol{f}$, an invertible function of $\boldsymbol{p}$, such that,

$$\boldsymbol{p} = \boldsymbol{h}(\boldsymbol{f}(\hat{\boldsymbol{\xi}})), \tag{9.18}$$

and we can state directly the required estimation function $\boldsymbol{h}$ as,

$$\boldsymbol{h} = \boldsymbol{f}^{-1}. \tag{9.19}$$

Thus we can calculate the estimate directly, *so we no longer need either search or minimisation routines and the major computational requirement of the system is removed.*

Furthermore, a relatively small training set can be used because the function $\boldsymbol{h}$ can interpolate between members in the training set. Clearly the system will give a more accurate interpolation with a large number of training images and it is important to note that the system has no extra computational or memory requirements during parameter estimation when extra training images are used. Since we know the form of $\boldsymbol{f}$, we do not need to store Ξ and there is no compromise between accuracy and speed. The extra computational requirements are restricted to the offline training process.

We also no longer have any restrictions on the training set, because we do not need to construct the high-dimensional surface for interpolation. We are therefore free to choose our training set such that problematic areas of parameter space are densely populated, and simple areas are sparsely populated.

Finally, since we can freely assign Ξ, we can separate the estimation of each of the u elements in $\boldsymbol{p}$ into u distinct problems. Let $\boldsymbol{\zeta}_i$ be the ith row of Ξ, which is the vector containing the ith feature for all of the training images. Now, if we choose Ξ such that,

$$\boldsymbol{\zeta}_i = f_i(g(p_j)), \tag{9.20}$$

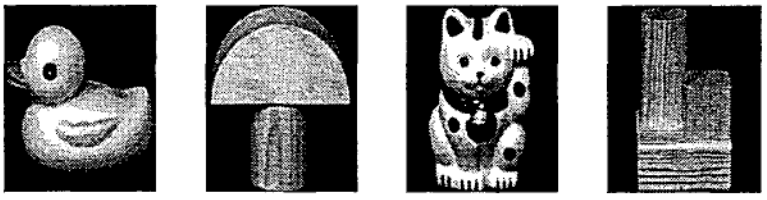


Figure 9.2: Toys from COIL database.

then each feature ζ_i is dependent on only one parameter and that parameter can be estimated independently of all others.

The simplest of all choices for Ξ is,

$$\Xi = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}, \tag{9.21}$$

In this case the feature extractor directly returns an estimate of the parameter vector $\boldsymbol{p}$ and the estimation function, $\boldsymbol{p} = \boldsymbol{h}(\hat{\boldsymbol{\xi}}) = \hat{\boldsymbol{\xi}}$, is trivial.

9.7 Examples

We now illustrate the significant advantages of explicit inversion by comparing its performance to the standard techniques on a number of image classification and parameter estimation problems.

9.7.1 Image Classification

The first example is an image classification task. We have four different toys, each of which can appear at any angle of rotation around the object's natural axis. Images of the four toys are taken from the COIL database [77], and can be seen in Figure 9.2. Our task is to classify a set of such images as belonging to two classes, labelled 'animal toy' and 'wooden toy', irrespective of the angle of the object.

We are given a training set of 24 correctly labelled images consisting of 6 images for each toy, rotated in 30° steps. The test set contains 228 images, 72 for each toy, rotated in 5° steps.

This is a challenging task, because the system is required to cope with multiple objects at multiple rotation angles belonging to the same class.

We have investigated this problem with all of the procedures described above, and the results can be seen in Tables 9.1 and 9.2. Each row in the tables corresponds to a particular feature extractor. Each column corresponds to a specific classifier. As stated previously, a compromise between speed and accuracy must be made when deciding on the number of prototypes to be used by each feature extractor. When selecting the values shown here, we put equal weight on the speed and accuracy of the results. In the case of Fisher-images the construction of the prototypes only allows

228 images		Number of Misclassifications						
Extractor	Features	nn	3-nn	5-nn	7-nn	ldf	Eq. 9.17	Eq. 9.16
correlation	24	14	29	34	74	3	na	na
synergetic	2	na	na	na	na	na	3	na
eigen-image	5	27	43	70	101	83	na	na
fisher-image	1	23	23	23	23	23	na	na
design (Eq. 9.15)	1	na	na	na	na	na	na	3

Table 9.1: Toy Classification Errors.

228 images		Classifier - Times (seconds)						
Extractor	Features	nn	3-nn	5-nn	7-nn	ldf	Eq. 9.17	Eq. 9.16
correlation	24	35.1	35.2	35.2	35.3	34.5	na	na
synergetic	2	na	na	na	na	na	0.81	na
eigen-image	5	4.47	4.55	4.62	4.67	4.02	na	na
fisher-image	1	0.78	0.85	0.91	0.99	0.42	na	na
design (Eq. 9.15)	1	na	na	na	na	na	na	0.41

Table 9.2: Toy Classification Times.

for a maximum of one class, as this is a two class problem. For our designed feature extractor, we used the design given in Equation 9.15.

Three different schemes in Table 9.1 returned the lowest error count of 3, or approximately 1.5%. It is interesting to note that the best choice between nearest-neighbour (nn), k-nearest neighbour (k-nn) and linear discriminant function (ldf) classifiers is dependent on the choice of feature extractor.

Of the three minimum-error options, the designed feature extractor is clearly the fastest, as seen in Table 9.2. The 24 prototypes required by the correlation method makes it 90 times slower than the proposed method. The synergetic feature extractor also performs well in terms of time requirements.

Table 9.2 makes it clear that most of the time requirement was spent extracting features, so the choice of classifier had only a small effect on the classification times. This is only true, however, with a small number of training images. As the size of the training set increases, the time used by the classifier will dominate the classification times. In comparison, the time taken by the direct classifier is small and dependent only on the number of prototypes. Fortunately, we can design the prototypes to keep their number to a minimum. In this 2-class case, in fact, the design of Equation 9.15 requires just a single prototype.

Figure 9.3 shows the values of $\hat{\xi}$ extracted from the test data by the designed feature extractor, with the class boundary given by Equation 9.16 shown as a solid line. The classification power of the extractor is made clear by the dashed line, which

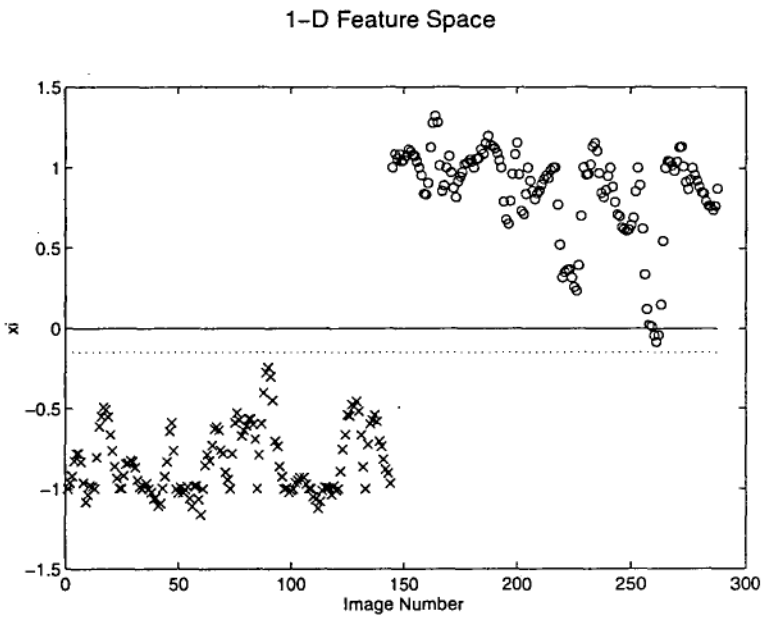


Figure 9.3: Toy classification test points are separable in the 1-D feature space.

has separated the classes completely.

9.7.2 Parameter Estimation

We now look at two examples of a parameter estimation problem, namely estimating the unknown pose, or rotation angle, of an object. Our object of interest is the rubber duck shown in Figure 9.2, and we use the same set of 72 images rotated in 5° steps described previously.

In the first of these experiments we train the system with a small number of images containing the object at a known pose and test the ability of the image analysis systems discussed above to interpolate values between images in the training set. In the second experiment, we train the system to act as a database containing a large number of training images and test the ability of the system to correctly recall the pose values in the presence of noise.

Given a set of training images with known pose θ, we must define Ξ in such a way that we can calculate the unknown pose $\hat{\theta}$ directly from $\hat{\xi}$. We choose,

$$\Xi = \begin{pmatrix} \sin(\theta + \phi) \\ \cos(\theta + \phi) \end{pmatrix}, \tag{9.22}$$

where ϕ is a constant angle which will be calculated below.

Figure 9.4 shows feature space populated by the training data. It is clear that the training points have been projected onto the unit circle at angles $\theta + \phi$, which is our smooth manifold prescribed in **C3b**. This example demonstrates the flexibility of the explicit inverse approach in that we have been able to use our knowledge that the parameters are periodic to design a smooth manifold.

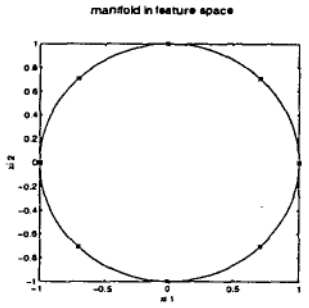


Figure 9.4: Pose interpolation training points in 2-D feature space.

Now as required, we clearly have an invertible function, such that given ϕ and $\hat{\boldsymbol{\xi}}$, we can calculate two different estimates for $\hat{\theta}$,

$$\theta = \tan^{-1}\left(\frac{\hat{\xi}_1}{\hat{\xi}_2}\right) - \phi, \tag{9.23}$$

$$\theta = \begin{cases} \sin^{-1}(\hat{\xi}_1) - \phi & \text{if } \hat{\xi}_2 \geq 0 \\ -\sin^{-1}(\hat{\xi}_1) + \frac{\pi}{2} - \phi & \text{if } \hat{\xi}_2 < 0. \end{cases} \tag{9.24}$$

In the case that all points were projected onto the circle, Equations 9.23 and 9.24 would be equivalent. This is not the case in general, however, and better results are achieved by using each formula in different situations, as explained in the examples below.

Another advantage of using the unit circle as our manifold, is that we know that whenever a test point is not projected directly onto the unit circle, there is an uncertainty involved in our estimation. So as well as directly calculating an estimate of the parameter, we can directly calculate a measure of the uncertainty involved,

$$\text{uncertainty} = |1 - \hat{\xi}_1^2 - \hat{\xi}_2^2|. \tag{9.25}$$

Interpolation

For the interpolation experiment, we selected 8 images in 45° steps as our training images.

Experience with curves and surfaces interpolated through the training points in feature space suggests that the smoothness of the interpolated surface increases with decreasing reconstruction error over the training set. This experience is also in keeping with criterion **C2**. While we have chosen the unit circle as our manifold, this can be parameterised by ϕ, so in fact we have a family of possible manifolds available to us. We have chosen ϕ to minimise the reconstruction error for prototype 1, and thereby increase the smoothness of the sine curve approximation. Equation 9.24 is the correct choice of formula for estimating the unknown pose because it interpolates solely on the sine curve, using the cosine curve only to resolve ambiguities.

Pose Interpolation			Errors and Times		
Feature Extractor	Features	Estimator	mean°	max°	seconds
correlation	8	minimisation	3.06	13.0	5.21
synergetic	8	minimisation	2.94	8.85	5.91
eigen-image	5	minimisation	3.21	18.72	3.24
design (Eq. 9.22)	2	Equation 9.24	3.67	9.64	0.29

Table 9.3: Pose Interpolation Results.

The pose estimation results and times can be seen in Table 9.3, where each element of the table represents a combination of a feature extractor with a parameter estimator. We used a minimisation-based estimator with the standard feature extractors because search-based estimators require a large training set, and are therefore unsuitable for the interpolation task at hand. All of the feature extractors returned mean error values within a single degree of each other. Comparing maximum error values, the best performing feature extractors were the synergetic extractor and the designed feature extractor.

It is in comparing calculation times that the advantages of the designed feature extractor/estimator pair become apparent. The explicit inversion process was one order of magnitude faster than its nearest competitor. Furthermore, this time comparison will become more favourable with larger problems. In Chapter 10, for example, we report on a situation where the pose estimation is two orders of magnitude faster.

Database Recall

As previously stated, the standard approaches to parameter estimation must all make a compromise between accuracy and time/memory requirements. In contrast, explicit inversion has the enviable ability to increase accuracy simply by adding new training images without any extra costs. To demonstrate this, we have trained the feature extractor using the complete set of 72 images. We are no longer interpolating between images, but rather rapidly recalling the instances that have already been learned.

The first row of Table 9.4 confirms that it correctly recalls all of the images in the same time and using the same memory requirements as were required for the interpolation example above. Therefore in the situation where training images can be inexpensively acquired, the optimum design strategy is to train the feature extractor with every available image.

To test the robustness of the recall, we have also added white noise to the test images before rescaling them to have unit length. Recalling that each training image has been normalised to have an amplitude of unity, we measure the strength of the white noise in terms of its amplitude. Table 9.4 shows the mean and maximum errors across the entire domain for a range of white noise amplitudes.

As we have added noise to the system, it is unlikely that either $\hat{\xi}_1$ and $\hat{\xi}_2$ will be better estimated than the other. In this situation Equation 9.23 with $\phi = 0$ is used,

Pose Database Recall			Errors		Times
Noise Amplitude	Features	Estimator	mean°	max°	seconds
0	2	Equation 9.23	0	0	0.28
0.1	2	Equation 9.23	0.06	0.23	0.28
0.2	2	Equation 9.23	0.11	0.42	0.28
1	2	Equation 9.23	0.19	1.98	0.28

Table 9.4: Pose Database Recall Results

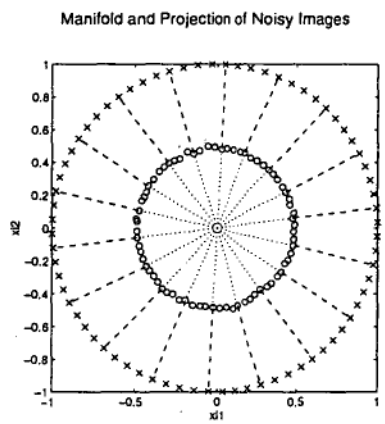


Figure 9.5: Pose database recall. The angle from the origin is robust to noise.

as the most robust estimator available.

Clearly our approach displays excellent robustness to white noise. The reason for this can be seen in Figure 9.5. Here we show the expected and actual pose manifolds when white noise with an amplitude of 1 is added to a one-dimensional case, trained with all possible images. This is the same system which produced the errors seen in the last row of Table 9.4. Note that the values of both $\hat{\xi}_1$ and $\hat{\xi}_2$ have been underestimated, but the angle from the origin to $\hat{\boldsymbol{\xi}}$ is relatively unaffected.

9.8 Conclusions

The explicit inversion approach to image analysis described here represents a fundamental shift from the mainstream. In creating a mapping from image space to parameter/class space, the standard approaches rely on inverting an unknown, possibly high-dimensional function. As this cannot be achieved directly, a number of schemes are commonly used to approximate the inversion. These techniques are typically either slow, or have poor classification power.

In contrast, we have mapped the image space to a known, low-dimensional function which can be inverted analytically. This results in an estimator which is both fast and powerful.

As the approach is object-independent, it can be followed for any image analysis

task. Furthermore, the technique can be applied to the general field of signal analysis, particularly when the input vectors are long. This is required such that a large number of training vectors can be used while maintaining the required linear independence.

There are a number of other advantages which come as a direct consequence of this change in approach. First, we can directly calculate an uncertainty for each estimate returned by the system. Second, as illustrated in Chapter 10, we can split a d-dimensional problem into d, 1-dimensional problems. Third, we can design our estimator to be robust against white noise. Fourth, the estimation function can be designed to take advantage of knowledge about the parameters available to the user. Fifth and finally, the memory and time requirements of the system are small and fixed. This means that we can increase the size of the training set without increasing time or memory costs, such that there is no compromise between accuracy and speed.

Chapter 10

Application to IR Jet Aircraft Pose Estimation

10.1 Motivation

Having established a new method of estimating the pose of an object, we now look at a particular application, namely that of tracking the pose of an aircraft.

The motivation behind this choice of application is a request from Australia's Defence Science and Technology Organisation (DSTO) to investigate the practicality of tracking the pose of a jet aircraft in real-time. We thank the DSTO for setting a challenge and for providing us with the data required to meet it.

The major application for such a system is in *tracking* the *location* of an aircraft. Rather than simply reacting to data on a frame-by-frame basis, tracking systems are generally built to predict the immediate future location of the aircraft. This prediction increases both the speed and robustness of the tracking. Current tracking systems base their prediction on the immediate past history of the aircraft's location. It is clear that the possible movements that an aircraft can make are highly constrained by the current pose of the aircraft, so supplementing the location information with pose information should lead to improved prediction and tracking.

A more advanced use of an aircraft pose estimation system is predicting *pilot intention*. As a simple example of this, an aircraft which is heading towards a target might well be considered to be intending to attack. Information about the pose of the aircraft, however, could support or question this conclusion, depending on whether the aircraft was approaching at an angle appropriate for an attack. An expert system which used both pose and location estimation to predict pilot intention could therefore be more accurate that one using location information alone.

There are three elements to 'practicality' when discussing the viability of a real-time pose tracking system. First is speed, second is invariance to translation and scale, and third is robustness against noise. We argue here that these three elements have been listed in descending order of importance. The speed of the estimation is of fundamental importance because of the nature of the aircraft's movement. A robust and accurate reading is of no use for tracking or predicting pilot intention if it has taken too long to produce. We have therefore concentrated our efforts on this first

element of speed, leaving invariance and robustness to future work.

10.2 The Problem

10.2.1 Estimating Pose and Tracking Pose

An aircraft does not stay still while we estimate its pose. In fact, given the incredible manoeuvrability of modern fighter aircraft, our goal is to track the pose of an aircraft faster than the 25 frames per second which is considered to be 'real-time' for most video applications.

While this requirement offers major challenges to pose estimation systems, it also offers one significant bonus. Given that an aircraft is an essentially rigid object with physical restrictions on its movements, there is undoubtedly a correlation between its pose at one instant and its pose at the next time step. Therefore, assuming that our pose estimate is correct at a given time, we can simply produce a reasonable estimate of where it will be at the next time step.

Figure 10.1 shows an illustrative subdomain of pose space with the current pose estimate in the centre, marked with a point. In a naive implementation of this idea, the area of pose space in which the next estimate is likely to be found forms a circle around the current estimate. This is shown by the solid line. A more sophisticated approach shown by the dashed ellipse, recognises that an aircraft can rotate around some axes faster than others and so the shape of the likely area must be altered accordingly.

A third, still more sophisticated model would also take advantage of the fact that the aircraft has *momentum*, and so any future changes in pose will be related to the immediate pose history. Thus our likely area should be decided based on the recent pose history of the aircraft, as marked with plus signs in Figure 10.1. This concept has been illustrated schematically using a dashed line in Figure 10.1.

We can take advantage of this knowledge to provide increased robustness to large errors. As a simple but important example, the symmetry of an aircraft ensures that many views 180° apart will be similar. Yet if pose is estimated frequently enough, it is not possible that an aircraft could rotate almost 180° between frames and so such an estimate could be ignored. More generally, we could improve our estimate of uncertainty to include some measure of the distance in pose space between our estimate and the likely area.

10.2.2 Infra-Red Imaging

The images for aircraft pose estimation are captured using sensors which are sensitive to the infra-red section of the electromagnetic spectrum. As such, the resulting image are significantly different to those captured using a standard video camera using the visible part of the spectrum.

Figure 10.2 shows two images of the same aircraft. Both of the images have been created artificially using a CAD model of the aircraft and rendering software. Figure 10.2(a) attempts to re-create an image taken in the visible spectrum, and uses a fixed light source behind the camera and the aircraft surface description to calculate the greyscale values. The light source is distant from the aircraft, such that the light

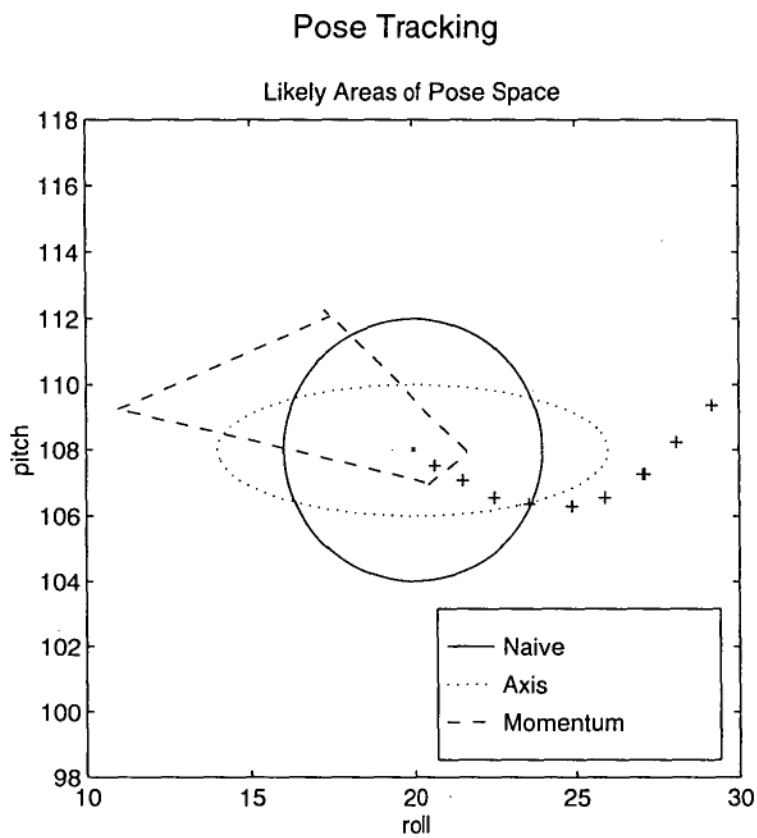
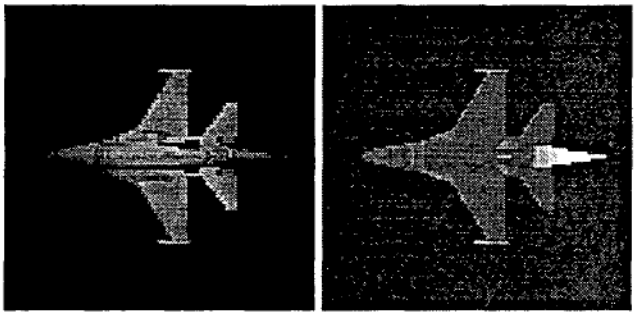


Figure 10.1: Increasingly sophisticated models of the likely pose at the next time step gives three different likely areas. The current pose is shown by a central point, and the immediate pose history by plus signs.



(a) (b)

Figure 10.2: Artificial visible and infra-red spectrum images.

rays are considered to be moving in parallel. Figure 10.2(b) attempts to re-create an image taken in the infra-red spectrum. There are no external light sources and each aircraft component has been assigned a self-luminosity according to the expected temperature of the component.

The most obvious distinction in the images is the flare of the jet engine. In the visible spectrum this is a very minor image feature, whereas the heat of the flare assures us that it is the dominant, bright feature in any infra-red image.

The most important distinction, and the reason why infra-red images are chosen for this task, is that the infra-red image does not contain any *shadow* effects. Shadow can significantly change the appearance of an object and is therefore problematic when our approach is based on the relationship between appearance and pose. In Figure 10.2(a) there are no extraneous objects such as clouds to cast a shadow on the image, but *self-shadowing* effects can clearly be seen. Thus in visible images, the direction of lighting will affect the appearance of the object and our pose estimate. Using infra-red imaging this problem is eliminated.

Clearly it would be advantageous to be able to test the algorithms with real images, but there are two reasons why this is difficult. First and foremost, the costs involved in acquiring such images makes it infeasible. Second, it is unclear how one would assign exact pose values to a real image. Given these two constraints, we have used artificial infra-red images for the work described below.

10.2.3 Dimensionality

An aircraft can rotate freely around any axis, so the task is to estimate three independent rotation angles for the aircraft. As described in Chapter 9, the explicit inversion approach to parameter estimation is simply scalable to any number of independent variables as we decouple the problem to estimate each independent variable separately. Unfortunately, the disk-space requirements to store enough images for the full three-dimensional problem are impractical, so we have demonstrated the efficacy of our approach in a subset of two dimensional space.

Figure 10.3 shows the relationship between the aircraft and the estimated pose

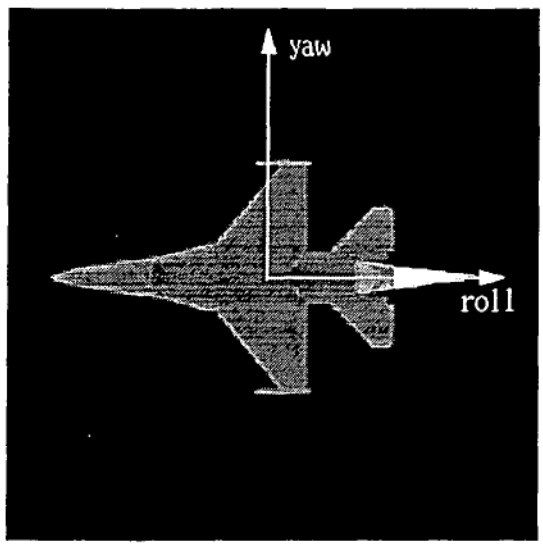


Figure 10.3: The two estimated pose variables.

variables. The definition of $(0, 0)$ in the pose variables is arbitrary.

10.2.4 The Dataset

The dataset consists of 1368 artificial images of a jet aircraft. Each image is square and has a side-length of 256 pixels. The aircraft rotates around the roll axis 90° in 5° steps. It rotates a complete revolution completely around the yaw axis.

10.2.5 Noise

There are a number of possible sources of white noise in a realistic aircraft pose estimation system. Among these are atmospheric conditions, radiation from nearby electronics and camera sampling. It is expected, however, that the noise levels experienced using high-quality video equipment, would be low.

More problematic for view-based pose estimation is correlated noise, or clutter, which is localised in regions of the image. This type of noise is caused by infra-red radiation emitted from objects other than the aircraft. The main source of this noise for a ground-based system would clearly be the sun, and to a lesser extent, temperature variations in the air. A pose estimation system located in an aircraft will have to cope with many sources of temperature variation. Among these are buildings, roads, rivers and coasts.

Strategies to remove this cluttered noise are a significant challenge themselves and are outside the scope of this work. They are currently being investigated in separate projects at DSTO. We therefore assume that pre-processing has removed most of the cluttered noise, but it is important that the pose estimation technique be as robust as possible to the noise that remains.

10.3 The Solution

The proposed solution for this problem is to use the explicit inversion technique described in Chapter 9. In that chapter we described symbolically how it was possible to *decouple* the pose parameters. Using the standard approach, each pose parameter is dependent of the entire feature vector. As a result of this, the pose parameter estimates are inter-dependent and each extra dimension adds new complexities. In contrast, explicit inversion allows us to simply define a separate interpolation curve for each parameter.

For this application with two pose variables, we can extend the one-dimensional case of Equation 9.22 to give,

$$\Xi = \begin{pmatrix} \sin(\theta_1) \\ \cos(\theta_1) \\ \sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix}, \tag{10.1}$$

and the pose parameters are estimated by

$$\hat{\theta_1} = \tan^{-1}\left(\frac{\hat{\xi}_1}{\hat{\xi}_2}\right), \quad \hat{\theta_2} = \tan^{-1}\left(\frac{\hat{\xi}_3}{\hat{\xi}_4}\right). \tag{10.2}$$

10.4 Examples

10.4.1 Aircraft Pose Estimation

Explicit Inversion vs Eigen-Image

This experiment demonstrates the speed and accuracy of explicit inversion in comparison to the standard technique. To be able to draw a fair comparison, we have used the same training set for both methods, not taking advantage of the database recall approach most suitable to explicit inversion. We used 84 images on a square grid covering the available pose space as the training set, and interpolated between these for all 1387 images in the test set. In Table 10.1, we show the errors for the explicit inversion process in comparison to an eigen-image/minimisation based routine. We also show the times required by the parameter estimator once the features have been extracted.

Clearly the most notable result here is the time requirements for explicit inversion, which is *two orders of magnitude faster* than the benchmark approach of Murase and Nayar. [73].

Effects of White Noise

In this experiment, we have used the optimum training method for explicit inversion, which is to use the largest possible training set. To test the robustness of the recall, we have also added white noise to the test images before rescaling them to have unit length. Recalling that each training image has been normalised to have an amplitude of unity, we measure the strength of the white noise in terms of its amplitude. Figure 10.4

F16	features	mean error	seconds
Eigen-Image	4	20.9°	52.3
Eigen-Image	6	11.61°	52.1
Eigen-Image	8	9.57°	54.6
Eigen-Image	10	9.10°	56.3
Direct	4	9.94°	.098

Table 10.1: Pose Estimation Errors.

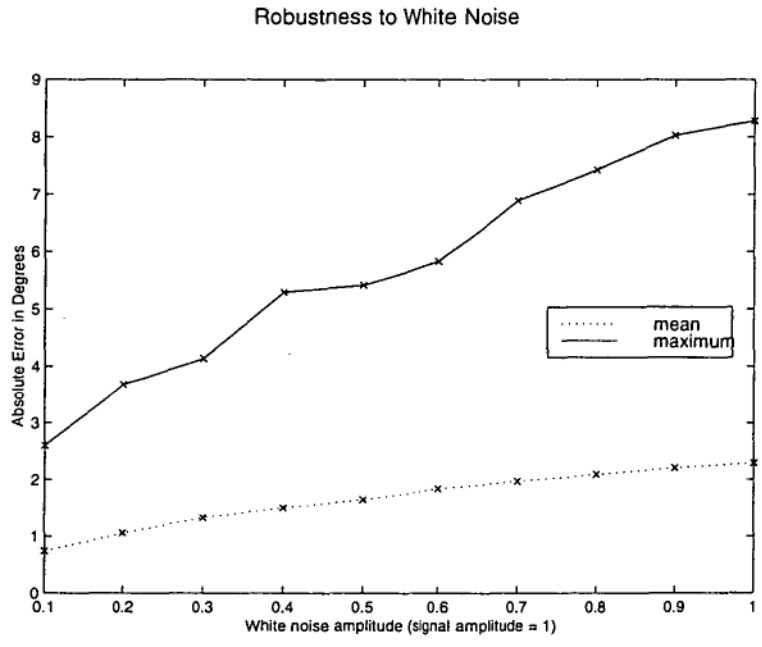


Figure 10.4: Explicit inversion has a natural robustness to white noise.

shows the mean and maximum errors across the entire domain for a range of white noise amplitudes.

This result confirms the finding of Chapter 9, that the explicit inversion approach displays excellent robustness to white noise.

Effects of Clutter

Without access to a more sophisticated model of clutter, we have taken a simplistic approach. For each input image we measure the total size of the aircraft in the image. We then create clutter consisting of a circle whose centre is coincident with a point on the aircraft and whose area is a given percentage of the size of the object in the original image. This percentage is our measurement of the size of the noise. The luminance value was constant over the circle and equal to the largest value in the original image.

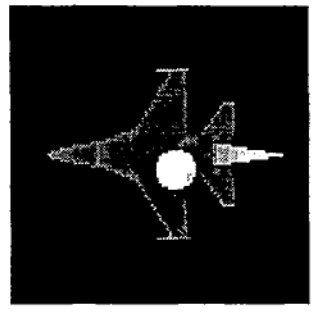


Figure 10.5: A typical infra-red image with randomly positioned clutter.

As with all input images, this input image including clutter is normalised to have unit length. Figure 10.5 shows how this simple clutter model affects a typical infra-red jet image when the clutter level is 10%. The same image without clutter is shown in Figure 10.2(b).

Figure 10.6 shows the results for our technique when clutter is added to each image.

Even with our simplistic clutter model, it is possible to conclude that our approach is susceptible to clutter in the image. This is not surprising, because our features are based on dot-products across the entire image, which are themselves susceptible to clutter. Recall from Chapter 9 that the competing view-based methods, such as the one proposed by Murase and Nayar [73], all construct their features in a similar way. They are therefore, also susceptible to clutter.

There are a number of strategies which could be used in parallel, in order to improve the system's robustness. The first is the use of sophisticated pre-processing to remove clutter. This could be based on the use of optical flow, and take advantage of a more sophisticated model of the likely clutter to be able to distinguish between the object and noise.

The second option is to use the uncertainty measure described above such that estimates with high uncertainty could be disregarded or treated with suspicion.

In practice, the measure of uncertainty associated with our circular pose manifold is of limited value. This is because our error is measured in degrees away from the correct angle, while uncertainty is measured as a Euclidean distance away from the manifold. In Figure 9.5, for example, all of the images are projected to a point approximately half way between the origin and the estimation manifold, so the uncertainty measure is quite high, but the error is low. This is a reasonable result using white noise as the amplitude of noise is in fact, high. But clutter may project an image into feature space with a radius near unity but at the wrong angle. In this case, the estimate will be considered to have low uncertainty, yet the error level is high.

10.4.2 Aircraft Pose Tracking

The Dataset

We do not have access to a dataset designed specifically to emulate any realistic aircraft motions. Yet as explained above, the correlation in pose between frames in a

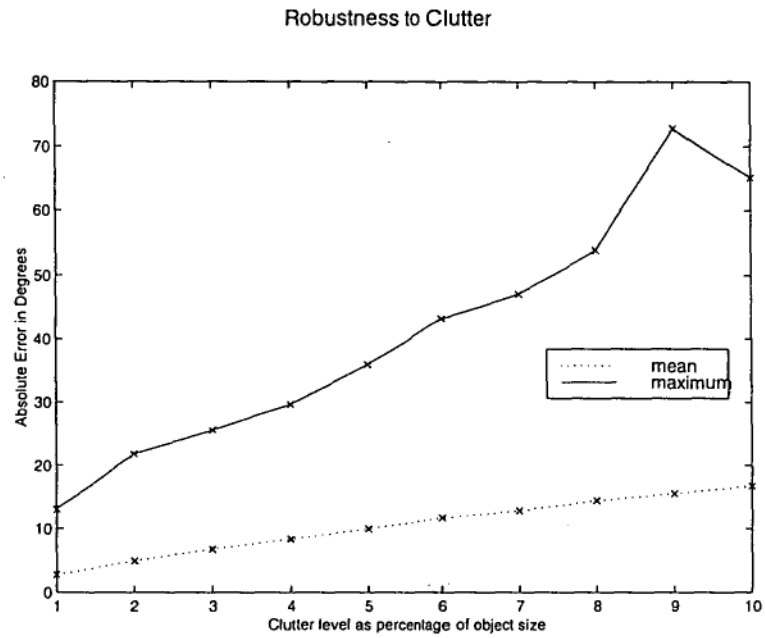


Figure 10.6: Explicit inversion is not robust against clutter.

continuous sequence of images is a valuable tool for adding robustness to the pose tracking system. We have therefore carried out some simple experiments with the dataset of still images described in Section 10.2.4. We created artificial trajectories by selecting arbitrarily chosen but reasonable flight sequences of still images.

Experiment: Tracking Pose

Figure 10.7 shows seven different artificial trajectories and compares them with the trajectory estimated by the system. These results did not use any higher level information such as the likely area for the next pose. Each of the images in the test trajectory had been corrupted with white noise of an amplitude equal to that of the original signal. This confirms visually the excellent robustness against white noise found in Table 9.4.

10.5 Conclusions

The goal of this work was to investigate the viability of view-based pose estimation for real-time infra-red pose tracking of aircraft. Methods reported in the literature are infeasible for this application because of *speed*. The manoeuvrability of modern jet aircraft is such that for a system to be valuable, it must be able to estimate pose at 25 frames per second or faster. On square images of side length 100 pixels, the current systems were only capable of half this speed or approximately 13 frames per second.

It became clear that until speeds faster than this became achievable, other questions concerning the viability of such systems were purely academic.

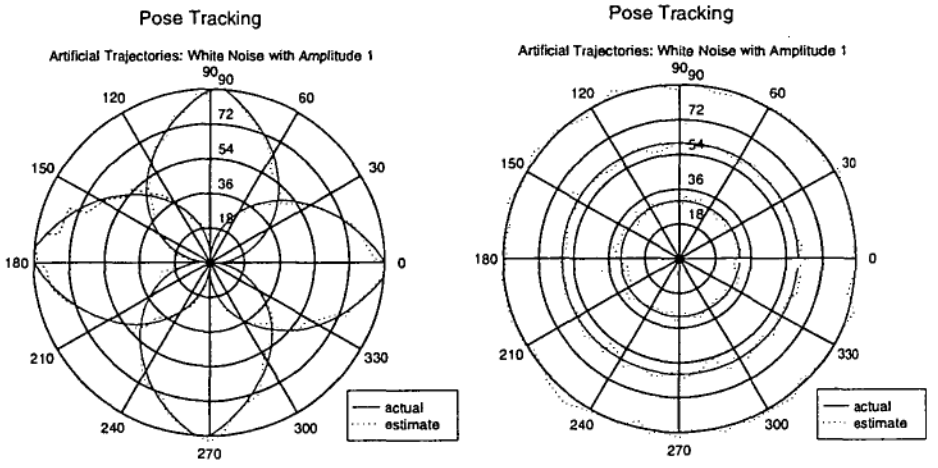


Figure 10.7: Tracking pose using artificial trajectories. These results used the optimum training method and corrupted the input images with white noise of amplitude equal to the original signal.

frames per second	feature extractor	parameter estimator
Standard	160	13
Explicit Inversion	160	2953

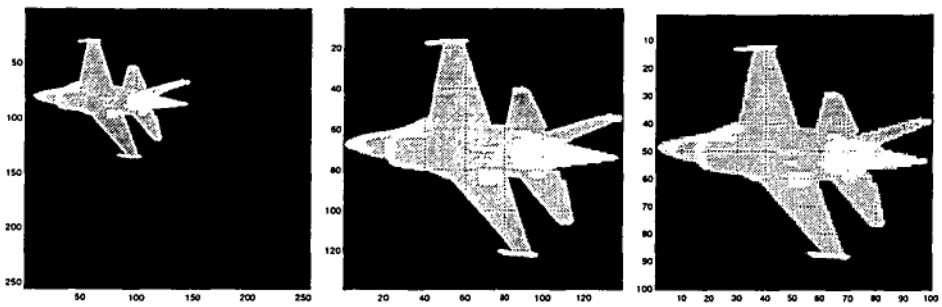
Table 10.2: Typical Frame Rates.

Our new method, known as 'explicit inversion' is an approach to view-based pose estimation which makes tracking aircraft in real-time a realistic possibility. Using this method we have fundamentally changed the structure of view-based pose estimation systems by completely removing the need for a search in feature space. On the same problem, our new approach is an order of magnitude faster, being capable of processing 160 frames per second.

It is a valuable exercise to break typical time costs over the two elements in the pose estimation system: feature extractor and parameter estimator. Table 10.2 shows the frames per second for these two elements on a Sun Ultra 1 UNIX box running uncompiled Matlab code.

It is clear that while for the standard approach, the system bottleneck is the parameter estimator, using explicit inversion, the system is now dependent on the time taken to extract the features. Currently the features are dot products of large vectors, so parallel processing could be used to speed the process even further.

The current rate of 160 frames per second is ample for this application.



(a) Original image.

(b) Remove translation and scale by selecting region of interest.

(c) Resample image to standard pixel dimensions.

Figure 10.8: Resampling an image to remove the effects of translation and scale.

10.5.1 Outstanding Issues

As argued in Section 10.1, the speed of the estimation for aircraft pose tracking is the fundamental issue. We now look briefly at the other important issues which are required for a practical pose tracking system.

Translation and Scale

The dataset used for these experiments contains artificial images where the aircraft is centred, and at a constant distance from the infra-red camera. Clearly, a practical pose tracking system would be required to cope with variation in these parameters.

There are three methods which could be used to address this issue. First is to estimate the values of these parameters and change the image input to the system to correct for the variation. For example, object-specific information such as the size and location of the jet flare in the image, could provide the required estimates. With this estimate, it would be possible to alter the image to correct for these variables accordingly.

A second, similar option is to re-sample all images, as shown in Figure 10.8. By treating all images in this way, translation ceases to be an issue. In terms of scale, the resampling will introduce a certain level of noise to the system. However, this noise will tend to have mean zero value and have a bell-shaped distribution, similar to that of white noise. This similarity in the noise distributions suggests that the system should be quite robust to the sampling noise.

Unfortunately, both of these options involve recalculating the input image, which is computationally expensive, due to the size of the image. It is therefore unlikely to be feasible for pose tracking.

The third option is to extract translation and scale independent features from the image. Promising results have been found by Fairney and Fairney [24], who have developed a set of features based on object edge points. Calculation of these features is, however, still computationally expensive, and the time requirements may make

real-time estimation problematic. Their feature extraction techniques are static, in that they do not use the current features as a starting point for extracting features in the next time step. Further work in creating a dynamic feature extractor may increase the viability of such features.

Clutter

The other major technical hurdle which must be faced before a pose estimation system can be of practical use for this application, is the issue of correlated noise, or clutter.

It is an undeniable truth that any view-based pose estimation system will face difficulties with this issue because the clutter can fundamentally change the image. There are two different approaches to this problem, both of which can be used in parallel, to maximise the robustness of the system to clutter.

The first is to pre-process the image to remove as much of the noise as possible. Again, object-specific information such as the likely temperatures of various aircraft components, along with techniques such as optical flow, could be used advantageously to remove clutter.

The second option is to calculate features which are more robust to clutter. The features used by Fairney and Fairney [24] show excellent robustness to both white and correlated noise. Unfortunately, these features are not directly suitable for use with the explicit inversion process due to the need for linear independence, as described in Chapter 9.

Further work in attempting to adapt these features for use within the explicit inversion technique, could lead to a practical, fast, and robust pose tracking system.

Chapter 11

Challenges and Conclusions

11.0.2 Challenges in Synergetic Image Analysis

The three major challenges facing synergetic image analysis are, as discovered in the example of aircraft pose estimation, invariance, robustness and speed. Our technique of explicit inversion answers the challenge of analysis speed by providing a fundamentally faster method of estimating image parameters, but the issues of invariance and robustness are still to be adequately addressed.

For example, humans are capable of recognising familiar faces irrespective of the style of haircut, the facial expression, the lighting or the amount of facial hair. Yet any one of these changes taken singly can prove to be very difficult for a computer-based face recognition system. Within reasonable tolerance zones, we also recognise objects invariantly with respect to rotation and viewing distance. Clearly, human performance is the main comparison by which we measure the success of computer-based image analysis, but it is interesting to see that our human understanding of robustness and invariance does not always match our strictly mathematical understanding. In Figure 11.1 for example, a human subject might quite reasonably state that the two images are the same, yet by looking at the images righted, we can see why the most basic of computer vision programs could state that they were different.

Perhaps the first step which must be tackled in achieving acceptable robustness

Figure 11.1: Human rotational invariance. From [40].

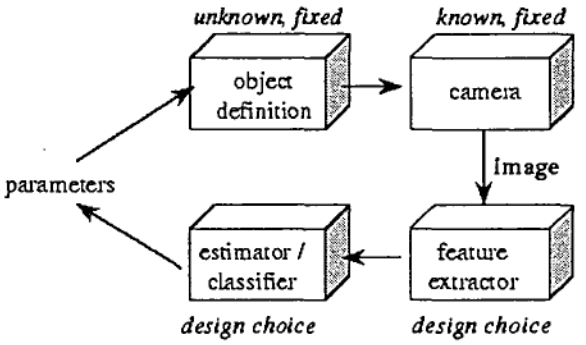


Figure 11.2: Schematic of an image analysis problem.

and invariance is to remove the *vectorisation* of images. With the exception of synergetic warping, all of our work has used a one-dimensional representation of images by digitising an image into pixels and raster-scanning them into column vectors, thereby throwing away valuable inter-pixel correlation information. Translation of an image, for example, is a simple operation when the pixels are in a two-dimensional matrix. In a column vector, it is not. A similar example is that of rotational invariance, which is generally treated by calculating a rotationally invariant transform of an image, and then vectorising the resulting signal. A good start has been made on two-dimensional synergetic pattern recognition by Yuasa et al. [107], but much work remains to be done.

The second step towards improved robustness and invariance is defining more sophisticated measures of similarity between images. Progress towards this goal should be greatly simplified by a successful two-dimensional image representation. In synergetic pattern recognition, for example, the dot product of vectorised images is the measure of similarity. Yet this measure has no concept of background or of a region of interest within the image, so all sections of the image are treated in the same way. Only with a more sophisticated concept of similarity will there be significant progress towards improving system robustness.

11.0.3 Conclusions

The goal of all image analysis is to complete the circuit shown in Figure 11.2. The goal of this work is to understand how natural synergetic systems can be used as a model to help complete this circuit.

The system shown in Figure 11.2 is perfect, in that the estimated parameters exactly equal the actual parameters. As described in Chapter 9, we have two controls for designing such a system. First is the design of the estimator/classifier. Second is the design of the feature extractor.

Part A: Synergetic Pattern Recognition and Learning

In Chapters 3, 4 and 5 of this work, we described three distinct elements of synergetic image analysis: recognition, rejection and learning. For each of these elements we learnt how to change either the classifier or feature extractor to increase the likelihood of completing the circuit of our ideal image analysis system.

The synergetic framework, through the central focus of the synergetic potential, allows us to construct a single process which combines recognition, learning and rejection.

The key to this understanding is the fact, as shown for the first time in this dissertation, that synergetic recognition, supervised learning, unsupervised learning and rejection, can all be expressed as the minimisation of a *single* synergetic potential.

Therefore, we can use the same process to learn, recognise and reject simultaneously. This confluence of learning and recognition parallels Haken's philosophy of synergetics, that pattern recognition *is* pattern formation (learning) [40].

This combination of traditionally distinct processes is more than simply aesthetically pleasing. First, since all of these processes are essentially special cases of the one larger process, improvements made in the synergetic potential for one of these processes, will have implications for the others. For example, in Chapters 3 and 4, we generalised the synergetic potential to allow for greater classification power and image rejection. This new potential holds the possibility of being used for learning more complicated classes as well as learning to reject spurious training images.

Second, as we have succeeded in unifying these three elements, it is worthwhile noting that the interesting challenge of including the *forgetting* of memories within this framework, is unexplored.

Finally, recall from Chapter 1 that synergetic pattern recognition is based on real physical systems, which leads to the possibility of synergetic hardware devices which perform pattern recognition. Now the unification of learning, forgetting, rejecting and recognising patterns, increases the possibilities for such devices.

In contrast, the traditional approach to pattern recognition, while capable of producing very powerful classifiers, has no such unity. Instead, each process is defined individually. Supervised learning, for example, alters the variables in the classifier to minimise classification error over the training set, while unsupervised learning attempts to minimise a function of the features, such as in Kohonen's self organising maps, which learn to cluster the feature sets together. Recognition and rejection, in contrast, are static processes, relying simply on the division of feature space given by the learning process.

Part B: Synergetic Pose Estimation

An important, and less often studied, sub-class of the general image analysis task shown in Figure 11.2, is continuous parameter estimation. In Part B of this dissertation we have investigated pose estimation, as an instance of parameter estimation. This is the first time that synergetic image analysis has been used for estimation of continuous parameters.

To achieve this, we have introduced a number of new techniques, each of which

attempt to complete the circuit of the ideal image analysis system by adapting either the estimator/classifier or the feature extractor, or both.

In Chapters 7 and 8 we follow the traditional line of thinking, where features are essentially fixed and the parameter estimator is changed so as to create the best possible agreement between the actual and estimated parameters. In essence, the fixed features leads to a fixed and sub-optimal learning process.

In contrast, Chapter 9 introduces explicit inversion, which creates problem-specific feature extractors and therefore, a flexible, tailored, supervised learning process. Because the learning process has been defined specifically to learn the task at hand, it is simple to create a parameter estimator that completes the circuit. As was shown in Chapters 9 and 10, using both the feature extractor and the parameter estimator controls to complete the circuit, leads to fundamentally superior parameter estimation systems.

Appendix A

Initial States Theorem

Proof: [Initial States Theorem] We do not know the exact location of the problematic region. We will therefore show instead that the theorem holds for a superset of the problematic region. This superset, which we label ϕ, is the hyper-rectangle with one vertex at ξ^*, going to positive infinity on all axes,

$$\phi = (\xi_i > \xi_i^*) \qquad \forall i = 1, \ldots, n. \tag{A.1}$$

Any image $\boldsymbol{q}$ can be decomposed so that

$$q = \sum_{i=1}^{n} r_i \boldsymbol{v}_i + \boldsymbol{w}, \tag{A.2}$$

where $\boldsymbol{w}$ is orthogonal to the prototypes,

$$\boldsymbol{w}\boldsymbol{v}_k^+ = 0, \qquad \forall k = 1, \ldots, n. \tag{A.3}$$

Now scaling $\boldsymbol{q}$ to have unit length and substituting into the definition of the order parameters (Equation 2.21), we find that

$$\xi_k = \frac{r_k}{|\sum_{i=1} n r^i \boldsymbol{v}_i + \boldsymbol{w}|}. \tag{A.4}$$

This is maximised when $\boldsymbol{w}$ is zero. Therefore, we can restrict our search for a test image $\boldsymbol{q}$ which will project into ϕ, to those which are linear superpositions of the prototypes, or $\boldsymbol{q} = \sum_{i=1}^{n} r_i \boldsymbol{v}_i$.

Now from the definition of ϕ (Equation A.1), if $\boldsymbol{q}$ is to be projected into ϕ, there are n inequality relationships which must hold simultaneously. Let us assume, without loss of generality, that the first $n-1$ of these inequalities are true,

$$\xi_i > \xi_i^* \qquad \forall i = 1, \ldots, n-1. \tag{A.5}$$

We now show that the nth inequality restriction cannot be true at the same time.

Squaring both sides and adding the left- and right-hand sides of these inequalities (Equation A.1), and using Equations (3.9) and (A.4), yields the inequality,

$$\frac{\sum_{i=1}^{n-1} r_i^2}{\sum_{i=1}^{n} r_i^2} > \frac{1}{c}\left[\frac{2(n-1)\sum_{i=1}^{n}\lambda_i}{2n-1} - \sum_{i=1}^{n-1}\lambda_i\right]. \tag{A.6}$$

Manipulation of this gives an inequality between the r_n and λ_n,

$$\frac{r_n^2}{\sum_{i=1}^{n} r_i^2} < \frac{1}{c}\left[\frac{c(2n-1)-s}{2n-1} + \lambda_n\right]. \tag{A.7}$$

This can then be re-expressed in terms of ξ_n and ξ_n^*,

$$\xi_n^2 < \xi_n^{*2} + \frac{1}{c}\left[c - \frac{\sum_{i=1}^{n}\lambda_i}{2n-1}\right]. \tag{A.8}$$

The final term can be re-expressed in terms of ξ^* using Equation 3.10, to give,

$$\xi_n^2 < \xi_n^{*2} + 1 - \sum_{i=1}^{n} \xi_i^{*2}. \tag{A.9}$$

Now setting

$$\sum_{i=1}^{n} \xi_i^{*2} \geq 1, \tag{A.10}$$

as given in the statement of the theorem, gives,

$$\xi_n^2 < \xi_n^{*2}. \tag{A.11}$$

This does not satisfy the requirements to be a member of ϕ as given by Equation A.1. Since q represents all possible images, we can conclude that no image can be projected into ϕ. ■

Appendix B

Avoiding the Singularity

At the center of the proposed feature extractor is the inversion of the correlation matrix Q^TQ. When it is singular, the system breaks down. However, the fact that the same form is used for both calculating G_p and extracting features from test images means that the Moore-Penrose pseudo-inverse will suffice. We replace Equation 9.13 with,

$$G_p = \Xi X = \Xi(Q^TQ)^+, \tag{B.1}$$

where the superscript plus denotes the pseudo-inverse. We now wish to confirm that the training values Ξ are recalled correctly using this new formulation. Substituting G_p back into Equation 9.6, we find,

$$\hat{\Xi} = \Xi X Q^T Q = G_p Q^T Q X Q^T Q \tag{B.2}$$

Now the pseudo-inverse has the property that $Q^TQXQ^TQ = Q^TQ$, so we can conclude that $\hat{\Xi} = G_pQ^TQ = \Xi$, as required.

Bibliography

[1] G. Anderson and W. Gaborsky, "The polynomial method augmented by supervised training for hand printed character recognition," in *Artificial neural networks and genetic algorithms*, (R. R. Albrecht, C. R. Reeves, and N. C. Steele, eds.), pp. 101–106, Springer-Verlag, 1993.

[2] M. Arbib and A. Hanson, *Vision, brain and cooperative computation.* Harvard: MIT Press, 1987.

[3] W. Banzhaf and H. Haken, "Learning in a Competitive Network," *Neural Networks*, vol. 3, pp. 423–435, 1990.

[4] E. Basar, H. Flohr, and H. Haken, eds., *Synergetics of the Brain.* Berlin: Springer-Verlag, 1983.

[5] D. T. M. Basin DT_NURBS documentation http://dtnet33-199.dt.navy.mil/dtnurbs/doc.htm.

[6] R. Basri and D. Weinshall, "Distance Metric Between 3D Models and 2D Images for Recognition and Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 465–470, 1996.

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[8] M. Bestehorn and H. Haken, "Associative Memory of a Dynamical System: The Example of Convection Instability," *Zeitschrift fur Physik B*, vol. 82, pp. 305–308, 1991.

[9] D. J. Beymer, "Face Recognition Under Varying Pose," Tech. Rep. 1461, MIT Artificial Intelligence Laboratory, 1994.

[10] M. J. Black and A. D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," in *Proceedings of ECCV'96*, (Cambridge), pp. 329–342, 1996.

[11] L. Blumenfeld, *Physics of Bioenergetic Processes.* Berlin: Springer-Verlag, 1983.

[12] F. G. Böbel and T. Wagner, "Theoretical Foundations of Synergetic Image Processing," in *Proceedings of the 1st International Conference of Applied Synergetics and Synergetic Engineering*, (Erlangen, Germany), pp. 46–52, June 1994.

[13] P. C. Bressloff, "Neural Networks, Lattice Instantons, and the Anti-Integrable Limit," *Physical Review Letters*, vol. 75, no. 5, pp. 962–965, 1995.

[14] J. Brochard, L. Coutin, and M. Leard, "Modelling of rigid objects by bidimensional moments. Applications to the estimation of 3D rotations.," *Pattern Recognition*, vol. 29, no. 6, pp. 889–902, 1996.

[15] R. Brunelli, "Estimation of Pose and Illuminant Direction for Face Processing," Tech. Rep. 1499, MIT Artificial Intelligence Laboratory, 1994.

[16] A. Daffertshofer and H. Haken, "A New Approach to Recognition of Deformed Patterns," *Pattern Recognition*, vol. 27, pp. 1697–1705, December 1994.

[17] A. Daffertshofer, H. Haken, W. Lorenz, and M. Ossig, "Hierarchical Structures in Pattern Recognition," in *Proceedings of the First International Conference on Applied Synergetic and Synergetic Engineering*, pp. 80–85, 1994.

[18] H. G. DeYoung and K. Ikeuchi, "Payday for Machine Vision," *Photonics Spectra*, vol. 28, pp. 76–84, 1994.

[19] M. Dhome, M. Richetin, J.-T. Laprestè, and G. Rives, "Determination of the Attitude of 3-D Objects from a Single Perspective View," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1265–1278, December 1989.

[20] D.Lehmann, D. Brandeis, H. Ozaki, and I. Pal, "Human Brain EEG Fields: Micro-states and Their Functional Significance," in *Computational Systems - Natural and Artificial*, (H. Haken, ed.), pp. 65–73, Springer-Verlag, 1987.

[21] R. A. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York City, NY: Wiley-Interscience, 1973.

[22] S. Dudani, K. Breeding, and R. McGhee, "Aircraft identification by moment invariants," *IEEE Transactions on Computers*, vol. 26, no. 1, pp. 39–46, 1977.

[23] N. Ezquerra and R. Mullick, "An Approach to 3D Pose Determination," *ACM Transactions on Graphics*, vol. 15, pp. 99–120, April 1996.

[24] P. T. Fairney and D. P. Fairney, "3-D object recognition and orientation from single noisy 2-D images," *Pattern Recognition Letters*, vol. 17, pp. 785–793, 1996.

[25] R. A. Fisher, "The Use of Multiple Measures in Taxonomic Problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.

[26] A. Fuchs, R. Friedrich, H. Haken, and D.Lehmann, "Spatio-Temporal Analysis of Multi-Channel Alpha EEG Map Series," in *Computational Systems - Natural and Artificial*, (H. Haken, ed.), pp. 74–83, Springer-Verlag, 1987.

[27] A. Fuchs, *Synergetische Systeme zur Mustererkennung und zur phänomenologischen Modellierung raum-zeitlich aufgelöst gemessener EEG's.* PhD thesis, University of Stuttgart, 1990.

[28] K. Fukushima, "Neocognitron: a hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, pp. 119–130, 1988.

[29] J. Gilbert and W. Yang, "A Real-Time Face Recognition System Using Custom VLSI Hardware," *Proceedings of the IEEE Workshop on Computer Architectures for Machine Perception*, pp. 58–66, 1993.

[30] Goldberg and Mason, "Bayesian Grasping," *Proceedings of the IEEE International Conference on Robotic Automation*, pp. 1264–1269, 1990.

[31] G. H. Golub, V. Klema, and G. W. Stewart, "Rank Degeneracy and Least Squares Problems," Tech. Rep. TR-456, Department of Computer Science, University of Maryland, 1976.

[32] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore: John Hopkins, 2 ed., 1989.

[33] N. Greene, "Transformation Identities," in *Graphics Gems*, (S. Glassner, ed.), pp. 485–493, Academic Press, 1990.

[34] K. D. Gremban and K. Ikeuchi, "Appearance-based vision and the automatic generation of object recognition code," in *Three-Dimensional Object Recognition Systems*, (A. Jain and P. Flynn, eds.), pp. 229–258, Elsevier Science Publishers, 1993.

[35] K. D. Gremban and K. Ikeuchi, "Planning Multiple Observations for Object Recognition," *International Journal of Computer Vision*, vol. 12, pp. 137–172, 1994.

[36] R. Haas, *Bewegungserkennung und Bewegungsanalyse mit dem Synergetischen Computer*. PhD thesis, University of Stuttgart, 1995.

[37] H. Haken, "Synergetic Hardware through Lasers and Semiconductors," in *Proceedings of the First International Conference on Applied Synergetics and Synergetic Engineering*, (F. G. Böbel and T. Wagner, ed.), pp. 183–193, 1994.

[38] H. Haken, *Synergetics. An Introduction*. Vol. 1 of *Springer Series Synergetics*, Berlin, Heidelberg: Springer-Verlag, 3 ed., 1983.

[39] H. Haken, *Advanced Synergetics. Instability Hierarchies of Self-Organising Systems and Devices*. Vol. 20 of *Springer Series Synergetics*, Berlin, Heidelberg: Springer-Verlag, 1987.

[40] H. Haken, *Synergetic Computers and Cognition*. Vol. 50 of *Springer Series Synergetics*, Berlin, Heidelberg: Springer-Verlag, 1991.

[41] H. Haken, R. Haas, and W. Banzhaf, "A New Learning Algorithm for Synergetic Computers," *Biological Cybernetics*, vol. 62, pp. 107–111, 1989.

[42] P. Hallinan, "A Low-Dimensional Representation of Human Faces for Arbitrary Lighting Conditions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 995–999, 1994.

[43] T. Hogg, D. Rees, and H. Talhami, "Three-dimensional pose from two-dimensional images: a novel approach using synergetic networks," in *Proceedings of the IEEE International Conference on Neural Networks '95*, pp. 1140–1144, December 1995.

[44] T. Hogg and H. Talhami, "A Competitive Non-Linear Approach to Object Recognition: The Generalised Synergetic Algorithm," in *Proceedings of the 4th Australian and New Zealand Conference on Intelligent Information Systems '96*, November 1996.

[45] T. Hogg, H. Talhami, and D. Rees, "A Practical Approach to View-Based Synergetic Pose Estimation," in *Proceedings of the IEEE Speech and Image Technologies for Computing and Telecommunications (TENCON)*, 1997.

[46] T. Hogg, H. Talhami, and D. Rees, "Explicit Inversion: An Approach to Image Analysis," *Pattern Recognition*, vol. , no. , p. , 1998. submitted.

[47] T. Hogg, H. Talhami, and D. Rees, "Tracking Pose Using View-Based Synergetic Recognition," in *Proceedings of the International Conference on Signal Processing and Communications (ICSPC)*, 1998.

[48] V. D. Hollard and J. D. Delius, "Rotational Invariance in Visual Pattern Recognition by Pigeons and Humans," *Science*, vol. 218, pp. 804–806, 1982.

[49] R. J. Holt and A. N. Netravali, "Uniqueness of Solutions to Three Perspective View Points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 303–307, 1995.

[50] R. Horaud, "New methods for matching 3-D objects with single perspective views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 3, pp. 401–412, 1987.

[51] H. Horner, "Dynamics of Spin Glasses and Related Models of Neural Networks," in *Computational Systems - Natural and Artificial*, (H. Haken, ed.), pp. 118–132, Springer-Verlag, 1987.

[52] T. Huang and C. Lee, "Motion and structure from orthographic projections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 536–540, 1989.

[53] S. V. Huffel and J. Vandewalle, "Subset Selection Using the Total Least Squares Approach in Collinearity Problems with Errors in the Variables," *Linear Algebra and Its Applications*, vol. 88/89, pp. 695–714, 1987.

[54] K. Ikeuchi and T. Kanade, "Automatic generation of object recognition programs," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 1016–1035, 1988.

[55] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis.* New Jersey: Prentice Hall, 3 ed., 1992.

[56] G. Kanizsa, *Organisation in vision: essays on gestalt perception.* New York: Praeger, 1979.

[57] E. Knobloch, A. E. Deane, and J. Toomre, "Oscillatory Doubly Diffusive Convection: Theory and Experiment," in *The Physics of Structure and Formation*, (W. Güttinger and G. Dangelmayr, eds.), pp. 117–129, Springer-Verlag, 1986.

[58] J. Koenderink and A. van Doorn, "The internal representation of solid shape with respect to vision.," *Biological Cybernetics*, vol. 32, pp. 211–216, 1979.

[59] T. Kohonen, *Associative memory: a system-theoretical approach.* Berlin: Springer-Verlag, 1977.

[60] T. Kohonen, *Self-Organisation and Associative Memory.* Vol. 8 of *Springer Series Information Sciences*, Berlin: Springer-Verlag, 1 ed., 1984.

[61] T. Kohonen, *Self-Organisation and Associative Memory.* Vol. 8 of *Springer Series Information Sciences*, Berlin: Springer-Verlag, 2 ed., 1987.

[62] A. Laurentini, "Efficiently computing and representing aspect graphs of polyhedral objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 57–58, January 1996.

[63] D. Lowe, "The viewpoint consistency constraint," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 57–72, 1987.

[64] G. M. T. Man, J. Poon, and W. Siu, "Pose Estimation for known arbitrary and noisy planar curves," *Electronics Letters*, vol. 31, no. 12, pp. 960–962, 1995.

[65] B. S. Manjunath, S. Chandrasekaran, and Y. F. Wang, "An Eigenspace Update Algorithm for Image Analysis," in *Proceedings of IEEE International Symposium on Computer Vision*, (Coral Gables, FL), November 1995.

[66] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *SIAM Journal*, vol. 11, pp. 431–441, June 1963.

[67] K. Matsuoka, "An associative network with cross inhibitory connections," *Biological Cybernetics*, vol. 61, pp. 393–399, 1989.

[68] B. Moghaddam and A. Pentland, "Probabalistic Visual Learning for Object Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 696–710, July 1997.

[69] R. Mukundan, "Estimation of quaternion parameters from two dimensional image moments.," *Graphical Models and Image Processing*, vol. 54, no. 4, pp. 345–350, 1992.

[70] R. Mukundan, N. Malik, and K. R. Ramakrishnan, "Attitude Estimation using moment invariants," *Pattern Recognition Letters*, vol. 14, pp. 199–205, 1993.

[71] R. Mukundan and K. R. Ramakrishnan, "An iterative solution for object pose parameters using image moments.," *Pattern Recognition Letters*, vol. 17, pp. 1279–1284, 1996.

[72] H. Murase and M. Lindenbaum, "Partial Eigenvalue Decomposition of Large Images Using Spatial Temporal Adaptive Method," *IEEE Transactions on Image Processing*, vol. 4, no. 5, pp. 620–629, 1995.

[73] H. Murase and S. K. Nayar, "Learning and Recognition of 3D Objects from Appearance," in *IEEE Qualitative Vision Workshop*, (New York, USA), pp. 39–50, June 1993.

[74] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *International Journal of Computer Vision*, vol. 14, pp. 5–24, 1995.

[75] S. K. Nayar and H. Murase, "Parametric Appearance Representation," in *Early Visual Learning*, (S. Nayar and T. Poggio, eds.), pp. 131–160, New York: Oxford University Press, 1996.

[76] S. A. Nene and S. K. Nayar, "A Simple Algorithm for High Dimensional Nearest Neighbour Search," Tech. Rep., Computer Science, Columbia University, 1995.

[77] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Tech. Rep., Computer Science, Columbia University, 1996.

[78] A. Noble, D. Wilson, and J. Ponce, "On computing aspect graphs of smooth shapes from volumetric data," *Computer Vision and Image Understanding*, vol. 66, pp. 179–192, February 1997.

[79] Y. Noguchi, "Subspace method of feature extraction using non-symmetric projection operators," *Bull. Ellectrotech. Lab. (Japan)*, vol. 40, pp. 571–587, 1976.

[80] J. Ostuni and S. Dunn, "Motion from Three Weak Perspective Images Using Image Rotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 64–69, 1996.

[81] H. Plantinga and C. R. Dyer, "Visibility, occlusion and the aspect graph," *International Journal of Computer Vision*, vol. 5, pp. 137–160, 1990.

[82] T. Poggio and D. Beymer, "Regularization Networks for Visual Learning," in *Early Visual Learning*, (S. Nayar and T. Poggio, eds.), pp. 43–66, New York: Oxford University Press, 1996.

[83] T. Poggio and S. Edelman, "A network that learns to recognise three-dimensional objects," *Nature*, vol. 343, pp. 263–266, 1990.

[84] T. Poggio and F. Girosi, "A Theory of Networks for Approximation and Learning," Tech. Rep. 1140, M.I.T. Artificial Intelligence Laboratory, July 1989.

[85] J. Ponce, A. Hoogs, and D. J. Kriegman, "On Using CAD Models to Compute the Pose of Curved 3D Objects," *CVGIP: Image Understanding*, vol. 55, pp. 184–197, 1992.

[86] G. Poulton, "Synergetic Computers - Alternative Derivation of Adjoint Prototypes," 1995. (private communication).

[87] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*. Cambridge, UK: Cambridge University Press, 1986.

[88] R. J. Prokop and A. P. Reeves, "A survey of moment based techniques for unoccluded object representation," *Graphical Models and Image Processing*, vol. 54, no. 5, pp. 438–460, 1992.

[89] A. Reeves, R. Prokop, S. Andrews, and F. Kuhl, "Three-dimensional shape analysis using moments and Fourier descriptors," in *Proceedings of the 7th International Conference on Pattern Recognition*, (Montreal, Canada), 1984.

[90] I. Rock, D. Wheeler, and L. Tudor, "Can we imagine how objects look from other viewpoints?," *Cognitive Psychology*, vol. 21, pp. 185–210, 1989.

[91] M. Schmutz and W. Banzhaf, "Robust Competitive Networks," *Physical Review A*, vol. 45, no. 6, pp. 4132–4145, 1992.

[92] M. Seibert and A. Waxman, "Adaptive 3-D Object Recognition from Multiple Views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 107–125, 1992.

[93] R. N. Shepard and J. Metzler, "Mental Rotation of Three-Dimensional Objects," *Science*, vol. 171, pp. 701–703, 1971.

[94] L. Sirovitch and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *Journal of the Optical Society of America A*, vol. 2, pp. 519–524, 1987.

[95] L. Stark, D. Eggert, and K. Bowyer, "Aspect Graphs and Nonlinear Optimization in 3-D Object Recognition," in *Proc. 2nd Int. Conf. on Computer Vision*, (Tampa, Florida), pp. 6–10, 1988.

[96] A. Tsukamoto, C.-W. Lee, and S. Tsuji, "Detection and Pose Estimation of Human Face with Multiple Model Images," *Image and Vision Computing*, vol. 12, pp. 487–498, October 1994.

[97] A. Turing, "On Computable Numbers, with an Application to the Entscheidung Problem," *Proceedings of London Mathematical Society*, vol. 42, pp. 230–265, 1936.

[98] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[99] M. Turk and A. Pentland, "Face Recognition using Eigenfaces," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.

[100] S. Ullman, *The Interpretation of Visual Motion.* Cambridge, MA: MIT Press, 1979.

[101] T. Wagner and F. G. Böbel, "Testing Synergetic Algorithms With Industrial Classfication Problems," *Neural Networks*, vol. 7, pp. 1313–1321, 1994.

[102] T. Wagner, U. Schramm, and F. G. Boebel, "Synergetic learning for unsupervised texture classification tasks," *PhysicaD*, vol. 80, pp. 140–150, 1995.

[103] F.-Y. Wang, P. J. A. Lever, and B. Pu, "A Robotic Vision System for Object Identification and Manipulation using Synergetic Pattern Recognition," *Robotics & Computer-Integrated Manufacturing*, vol. 10, pp. 445–459, 1993.

[104] A. H. Watt, *Fundamentals of three-dimensional computer graphics.* Wokingham, England: Addison Wesley, 1989.

[105] D. Weinshall, M. Werman, and N. Tishby, "Stability and likelihood of views of three dimensional objects," in *Proceedings of the 3rd ECCV*, (Stockholm, Sweden), pp. 24–35, 1994.

[106] G. Weisbuch, "Patterns in Random Boolean Nets," in *The Physics of Structure and Formation*, (W. Güttinger and G. Dangelmayr, eds.), pp. 52–67, Springer-Verlag, 1986.

[107] H. Yuasa, S. Ito, K. Ito, and M. Ito, "Associative memory with the reaction-diffusion equation," *Biological Cybernetics*, vol. 76, pp. 129–137, 1997.

[108] A. Yudashkin, "A Topological Approach to the Pattern Classification in Neural Networks," in *Proceedings of ICNN'96*, (Washington, DC), 1996.